Universitatea Tehnica “Gh. Asachi” Iasi
Facultatea de Electronica si Telecomunicatii

Teza de doctorat

Contributii la Studiul Retelelor Celulare Neurale

Îndrumator stiintific: prof. Liviu Goras

Candidat: ing. Tiberiu-Dinu Teodorescu

IASI, 2002

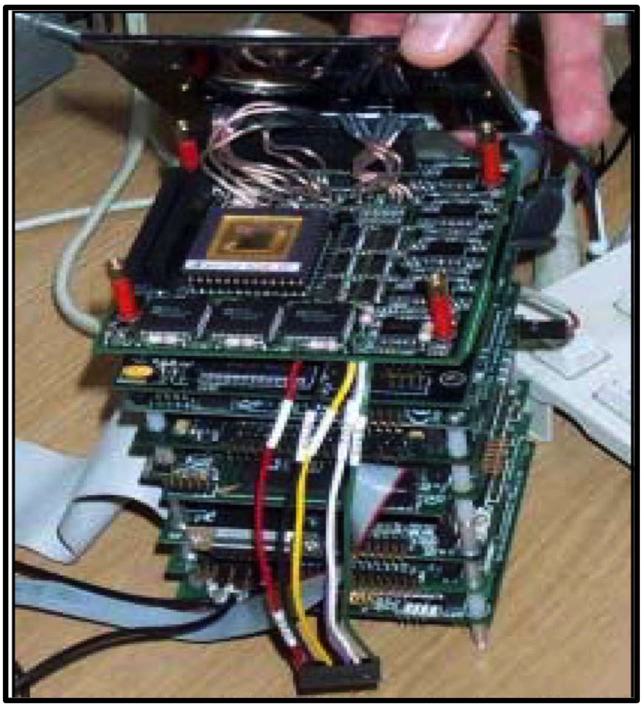
Introducere

În ultimii ani a luat o amploare deosebita notiunea de calculator analogic, bazat pe retele neuronale celulare (CNN[1]). O proprietate particulara a CNN-urilor fata de retelele neuronale clasice este aceea ca sunt foarte atractive pentru implementarea în tehnologie VLSI. Legaturile între entitatile componente sunt în acest caz locale, spre deosebire de retelele neuronale clasice, la care legaturile între entitatile componente sunt în cazul general totale[2]. Notiunea de functie de activare din teoria retelelor neuronale clasice[3] .este prezenta si la CNN-uri.
Pornind de la primele articole de Chua si Yang [1,2] în 1988, conceptul a fost dezvoltat de-a lungul timpului. Potentialul mare al acestui tip de sisteme vine din faptul ca este utilizabil într-o clasa de aplicatii relativ larga, pornind de la prelucrarea de imagini, conversia analog – numerica [3], generarea de semnal haotic [4-8], realizând tipurile de procesare enuntate la o viteza mult mai mare decât cea a unui calculator numeric performant, chiar realizat într-o arhitectura multiprocesor. Viteza acestui tip de sistem este evaluata în termeni de operatii pe secunda a 10 terra [9-15].
În prezent exista variante diverse de implementare a acestor sisteme, insa cea mai semnificativa este realizata de Institutul de Microelectronica de la Sevilla, Spania, care a realizat pâna în prezent chip-ul cu 128x128 de celule identic si local interconectate[4] [16,17]. Dupa realizarea chip-ului care implementeaza sistemul s-a trecut la valorificarea mai buna a posibilitatilor de procesare prin introducerea conceptului de Masina Universala de Calcul [18-27]. Pe scurt, implementarea acestui concept a permis folosirea sistemului pentru procesari mai complicate decât cele care se pot efectua într-un singur ciclu masina.
Câteva date despre procesorul analogic realizat în toamna lui 2001 ce foloseste conceptul de CNN sunt date mai jos:

Tehnologie	STM 0.35µm, 5M-1P
Stilul de design	Full custom pentru partea de nucleu de procesare si utilizând celule standard pentru partea de intrare/iesire digitala
Capsula	Ceramic QFP144
Numar de celule	16384 (retea de 128x128 celule)
Numar de tranzistoare	3.748.170
Tranzistoare/celula	198
Dimensiunea celulei	75.7µm x 73.3 µm
Densitatea celulelor	180 celule/mm^2
Gama dinamica a starii	0.6-1.4 V (programabila)
Gama dinamica a „ponderilor”	2.15-2.95 V (programabila)
Tactul pentru operatiile de I/O	32MHz

[1] CNN reprezinta acronimul termenului *Cellular Neural Networks*, tradus prin Retele Neuronale Celulare
[2] Legaturile sunt realizate în cazul retelelor neuronale clasice prin intermediul asa-numitelor ponderi. Acestea realizeaza conectarea unei celule cu toate celelalte, urmând ca, în functie de problema studiata, unele dintre ele sa se dovedeasca inutile (nu se modifica în timpul procesului de antrenare)
[3] Functia de activare este în cazul retelelor neuronale celulare de tipul amplificare cu saturatie.
[4] Timpul mediu de procesare a unei imagini de catre acest chip este de 160 ns.

Tensiunea de alimentare	3.3V (+/- 10%)
Putere consumata	<4 Watts
Numarul instructiunilor analogice	32
Numarul maxim al instructiunilor digitale	64 x 64 configuratii (pentru ponderi)
Dimensiune	11885 μm x 12230 μm



Numele de cod al procesorului analogic realizat este ACE16k si functioneaza la o rezolutie modesta, de 8 biti. Chiar daca acesta este un procesor analogic (toate operatiile au loc în domeniu analogic), el poate fi comandat dintr-un mediu digital. Pentru aceasta, procesorul încorporeaza un banc de convertoare D/A pentru furnizarea intrarii si un banc de convertoare A/D pentru furnizarea iesirii convertita în cod binar. În acest caz imaginea procesata este manipulata linie cu linie. Procesorul este proiectat sa lucreze atât cu imagini achizitionate de modulul optic al chip-ului (intrare analogica) sau sa fie interfatat cu un sistem digital. În ultimul caz lucreaza ca un co-procesor dedicat pentru prelucrari de imagini ultra-rapide. În fotografia alaturata se prezinta prototipul sistemului integrat (ACE Box) pilotat de procesorul analogic ACE4k si interfatat cu calculatorul. Procesorul ACE4k poate fi înlocuit cu cel din generatia urmatoare (ACE16k) si sistemul functioneaza cu modificari minore.

Teza de fata studiaza posibilitatile de procesare liniara si neliniara de semnal 1D si 2D, acoperind la “capitolul” de contributii originale o parte din clasa de aplicatii legate de prelucrarea de imagini si punând în evidenta pentru domeniul de functionare liniara a sistemului potentialul de procesare al retelelor celulare neuronale omogene.

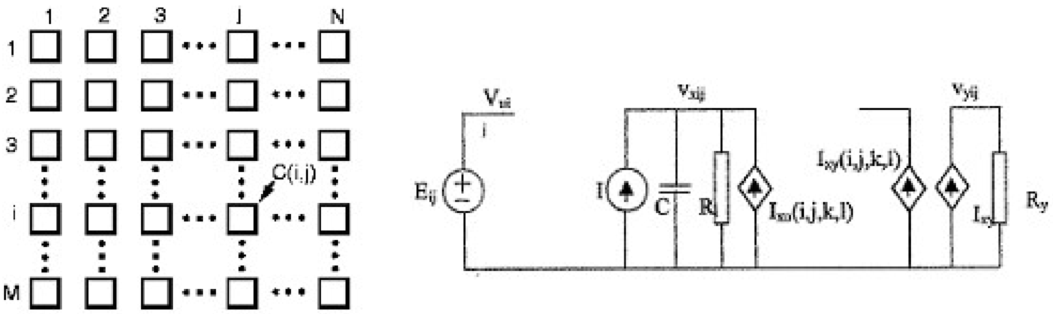
Cu toate ca multe dintre conceptele prezentate în teza îsi regasesc implementarea în realizarile actuale de circuit (mai ales cele prezentate în capitolele de analiza a stadiului actual al cercetarilor), aceasta nu îsi propune sa faca un studiu al posibilitatilor de procesare *strict* cu sistemele implementate în acest moment, ci îsi propune sa largeasca orizontul pentru analiza acestui tip de sisteme si proiectarea de instructiuni analogice asociate cu diverse tipuri de procesari de imagini. O posibila alternativa de folosire a rezultatelor teoretice obtinute consta în utilizarea lor la implementarea software a unor algoritmi bazati pe diverse structuri (realizate la nivel sistemic) a retelelor neuronale celulare.

Spre exemplu, una din întrebarile la care lucrarea de fata încearca sa dea un raspuns este daca, pentru realizarea unor procesari mai complicate de semnal, este necesar sa se mareasca aria de influenta directa asupra vecinilor a unitatii de procesare sau este mai eficient sa se mareasca ordinul de complexitate al unitatii de procesare? Pornind de la avantajul primordial al retelelor neuronale celulare fata de cele clasice, acesta se diminueaza odata cu cresterea ariei de influenta asupra vecinilor unei unitati de procesare.
Pe de alta parte, la cresterea complexitatii unitatii de procesare (celulei) numarul de tranzistoare necesare creste si prin urmare puterea consumata, aria si toti ceilalti parametrii prezentati mai sus se înrautatesc.
O alta abordare a unei procesari mai complicate de imagine este ca ea sa fie împartita functional în procesari simple, realizabile cu o celula si o arie de influenta directa a acesteia cât mai simple. Prin urmare si chip-ul necesar pentru aceste tipuri de procesari este eficient implementabil în tehnologie VLSI. Printr-o succesiune de procesari de semnal simple se poate obtine o procesare de semnal mai complicata.
La nivel de interfata software, succesiunea unor procesari simple efectuate asupra unei imagini. carora le corespund instructiuni analogice este de fapt un program realizat cu instructiuni analogice, ce se supune modelului de programare von Neumann. Din acest moment, odata stabilita structura chip-ului si a mediului integrat, folosirea acestora pentru realizarea unor procesari mai complicate de imagine, cum ar fi segmentarea orientata obiect se realizeaza prin scrierea unui program în limbaj Alpha (limbajul de nivel înalt proiectat special pentru acest tip de sisteme).

Capitolul 1.: Notatii de baza, definitii si teoreme semnificative

*Definitia 1****: Arhitectura CNN standard***

Arhitectura standard CNN [12] este constituita dintr-o matrice dreptunghiulara (MxN) de celule (unitati elementare de procesare), interconectate între ele prin intermediul *template*[1]-urilor (Fig. 1).



a) b)

Fig. 1: a) Arhitectura CNN standard; b) O realizare standard de circuit pentru unitatea de procesare

*Definitia 2****: Sfera de influenta a celulei C(i,j)***

Sfera de influenta Sr(i,j), [12] de raza r a celulei C(i,j) este definita ca multimea tuturor celulelor vecine care satisfac proprietatea:

$$S_r(i,j) = \{C(k,l) \mid \max_{1\le k\le M, 1\le l\le N} \{|k-i|, |l-j|\} \le r\} \qquad (1.1.)$$

în care r este un întreg pozitiv.

Ne vom mai referi la Sr(i,j) ca la o vecinatate (2r+1)x(2r+1). De exemplu, Fig. 2a) prezinta o vecinatate r=1 (3x3). În Fig. 2b) se prezinta o vecinatate r=2 (5x5).

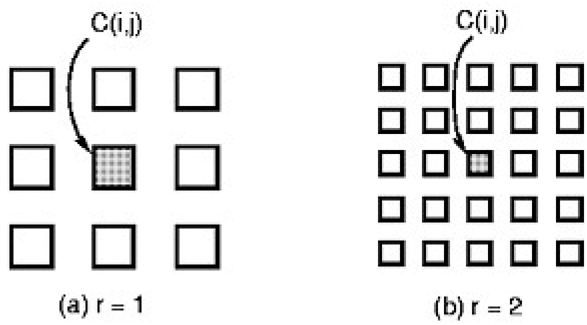


Fig. 2: a) Vecinatate simpla; b) Vecinatate dubla

[1] Se va folosi în continuare termenul provenit din literatura de specialitate anglo-saxona pentru evitarea confuziilor. *Template*-ul reprezinta expresia matematica a influentei starilor si intrarilor corespunzatoare celulelor vecine asupra starii si intrarii corespunzatoare celulei curente.

Definitia 3: ***Celule obisnuite. Celule de frontiera***

O celula C(i,j) este numita celula obisnuita [12] dupa Sr(i,j) daca toate celulele vecine $C(k,l) \in Sr(i,j)$ exista. În caz contrar, C(i,j) este numita celula de frontiera.

Definitia 4: ***CNN-ul standard***

Clasa CNN-ului standard MxN [12] este definita de o matrice dreptunghiulara de celule C(i,j), unde indicii reprezinta pozitia celulei în cadrul matricii; i=1..M, j=1..N. Fiecare celula este descrisa matematic de relatiile:

Ecuatia de stare

$$\dot{x}_{ij} = -x_{ij} + \sum_{C(k,l) \in Sr(i,j)} A(i,j;k,l) y_{kl} + \sum_{C(k,l) \in Sr(i,j)} B(i,j;k,l) u_{kl} + z_{ij} \tag{1.2.}$$

în care $x_{i,j} \in R$, $y_{kl} \in R$, $u_{kl} \in R$ si $z_{ij} \in R$ sunt denumite starea, iesirea, intrarea si pragul celulei C(i,j), iar A(i,j;k,l) si B(i,j;k,l) sunt denumite template-uri.

Ecuatia de iesire:

$$y_{ij} = f(x_{ij}) = \frac{1}{2} | x_{ij} + 1 | + \frac{1}{2} | x_{ij} - 1 | \tag{1.3.}$$

Aceasta este denumita neliniaritatea standard [12]. În esenta, scopul ei este de a limita semnalele de la iesirea celulelor si de a realiza, în anumite cazuri "binarizarea" imaginilor dupa procesare:

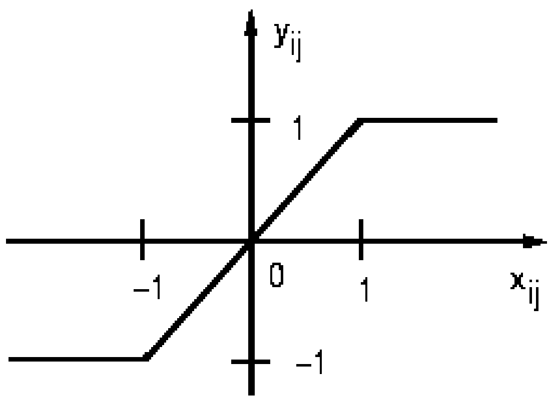


Fig. 3: Neliniaritatea standard

Cele doua ecuatii pun în evidenta faptul ca fiecare celula este influentata de iesirile celulelor vecine si de intrarile corespunzatoare celulelor vecine, prin intermediul template-urilor A si respectiv B.

Conditii de granita:

Conditiile de granita sunt date de comportarea unor celule din afara retelei de MxN celule parti *utile* în procesare.

Stare initiala:

$$x_{ij}(0), i = 1,...,M, j = 1,...,N \qquad (1.4.)$$

Observatii:

- intrarea u_{kl} este în mod obisnuit intensitatea pixelului unei imagini în tonuri de gri de dimensiune MxN, normalizata ($-1 = u_{kl} = 1$), unde albul este codat prin –1 iar negrul este codat prin 1. Pentru o imagine statica u_{kl} este constant, iar pentru o imagine în miscare u_{kl} este o functie de timp. Celelalte variabile (x(0),y,z) pot fi deasemenea imagini.
- în cazul cel mai general, A(i,j;k,l), B(i,j;k,l) si z_{ij} pot sa varieze cu pozitia în retea (i si ,j), dar si în timp. Daca nu este specificat în mod explicit acest lucru vom presupune ca sunt invariante în timp si în spatiu.
- în cel mai general caz A(i,j;k,l) si B(i,j;k,l) sunt operatori neliniari care actioneaza asupra lui $x_{kl}(t)$, $y_{kl}(t)$, $u_{kl}(t)$, $x_{ij}(t)$, $y_{ij}(t)$ si $u_{ij}(t)$, $0 = t = t_0$ si furnizeaza un scalar $(A(i,j;k,l)*y_{kl})(t_0)$ si $(B(i,j;k,l)*y_{kl})(t_0)$, $0 = t = t_0$.
- se pot introduce legi sinaptice dependente de stari (template-uri C) si de variabile amestecate (template-uri D). Acestea sunt $(C(i,j;k,l)*x_{kl})(t_0)$ si $(D(i,j;k,l)*(y_{kl},x_{kl})(t_0)$.

Este util sa ne gândim la tripletul {A(i,j;k,l), B(i,j;k,l), z_{ij}} ca la un program elementar de simulare a CNN-urilor (o instructiune analogica) , pentru ca el descrie modul în care o imagine U sau chiar doua (U si Y), de dimensiune MxN la t=0, va fi transformata pentru a produce o imagine Y(t) (de iesire) de dimensiune MxN pentru t>0. Dupa un timp în general foarte scurt imaginea de iesire se stabilizeaza (în ipoteza ca sunt îndeplinite anumite conditii de stabilitate).

*Definitia 5**: CNN-ul invariant în spatiu (isotrop)***

Un CNN este invariant în spatiu [12] daca si numai daca operatorii-template si z_{ij} nu variaza în raport cu spatiul. În acest caz, se poate scrie:

$$\sum_{C(k,l)\in Sr(i,j)} A(i,j;k,l)y_{kl} = \sum_{|k-i|\le r}\ \sum_{|l-j|\le r} A(i-k,j-l)y_{kl}$$

$$\sum_{C(k,l)\in Sr(i,j)} B(i,j;k,l)u_{kl} = \sum_{|k-i|\le r}\ \sum_{|l-j|\le r} B(i-k,j-l)u_{kl} \qquad (1.5.)$$

$$z_{ij} = z$$

Un CNN de clasa 2 standard [12] (cu template-uri liniare) are urmatoarea ecuatie de stare:

$$\dot{x}_{ij} = -x_{ij} + \sum_{C(k,l)\in Sr(i,j)} A(i,j;k,l)y_{kl} + \sum_{C(k,l)\in Sr(i,j)} B(i,j;k,l)u_{kl} + z_{ij}$$
$$+ \sum_{C(k,l)\in Sr(i,j)} C(i,j;k,l)x_{kl} + \sum_{C(k,l)\in Sr(i,j)} D(i,j;k,l)(u_{kl},x_{kl},y_{kl}) \quad (1.6.)$$

Existenta si unicitatea solutiei

Rezultate pregatitoare:

Teorema lui Peano (teorema de existenta a solutiei)
Fie h(t,x) o functie continua într-un dreptunghi R={(t,x), $|t-t_0|=a$, $|x-x_0|=b$}. Se definesc: **M=max{h(t,x)}** pentru t si x apartinând interiorului dreptunghiului si **a=min(a, b/M)**.
În aceste conditii problema Cauchy **dx/dt=h(t,x), $x(t_0)=x_0$** are solutie în intervalul **$I=[t_0-a, t_0-a]$**.

Teorema lui Picard-Lindelof (teorema de unicitate a solutiei)
Daca în plus fata de ipotezele enuntate în teorema de existenta, h satiface si conditia Lipschitz în x, atunci u(t) este unic.

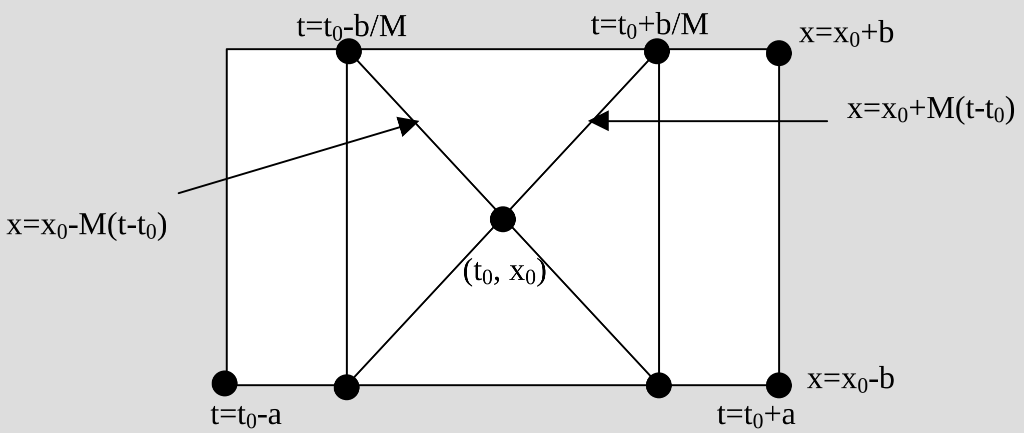


Fig. 4: Domeniul R pentru teoremele de existenta si unicitate

Liniile diagonale si dreptunghiul mai mic ilustreaza necesitatea de a restrictiona intervalul de timp în cazul în care M este destul de mare pentru ca b/M<a.

Ipoteze folosite pentru CNN-ul clasic:

- Operatorii sinaptici sunt liniari si fara memorie
- Intrarea $u_{ij}(t)$ si pragul (dat de sursele de curent) $z_{ij}(t)$ sunt functii continue de timp;

- Neliniaritatea f(x) este continua în sens Lipschitzian adica are proprietatea ca exista o constanta L în asa fel încât pentru orice x’ si x” apartinând multimii numerelor reale,

$$|f(x')-f(x'')|\le L\,|x'-x''|$$

Teorema 1
În aceste ipoteze, pentru orice stare initiala $x_{ij}(0)$ apartinând lui R, CNN-ul are o solutie unica, pentru orice t>0.

Demonstratie:
Utilizând ipoteza 1 se poate pune sistemul de ecuatii ce descrie CNN-ul în forma vectoriala de mai jos:

$$\dot{\mathbf{x}} = -\mathbf{x} + \hat{\mathbf{A}}\mathbf{y} + \hat{\mathbf{B}}\mathbf{u}(t) + \mathbf{z}(t) = \mathbf{h}(\mathbf{x}, t) \qquad (1.7.)$$

în care $y_i=f(x_i)$.
Utilizând ipoteza 3 se arata întâi ca h(x,t) este continua în sens Lipschitz ca functie de x. Se aleg x’ si x” apartinând lui R si se calculeaza y'=f(x') si y''=f(x'').

$$\begin{aligned} &\| \mathbf{h}(\mathbf{x}', t) - \mathbf{h}(\mathbf{x}'', t) \| = \| -\mathbf{x}' + \hat{\mathbf{A}}\mathbf{y}' + \mathbf{x}'' - \hat{\mathbf{A}}\mathbf{y}'' \| \\ &= \| \mathbf{x}'' - \mathbf{x}' + \hat{\mathbf{A}}(\mathbf{y}' - \mathbf{y}'') \| \\ &\le \| \mathbf{x}'' - \mathbf{x}' \| + \| \hat{\mathbf{A}} \| \, \| \mathbf{y}' - \mathbf{y}'' \| \end{aligned} \qquad (1.8.)$$

Se detaliaza termenul ce contine variabila y:

$$\begin{aligned} \| \mathbf{y}' - \mathbf{y}'' \| &= \left\| \begin{bmatrix} f(x_1^{'}) \\ f(x_2^{'}) \\ \vdots \\ f(x_n^{'}) \end{bmatrix} - \begin{bmatrix} f(x_1^{''}) \\ f(x_2^{''}) \\ \vdots \\ f(x_n^{''}) \end{bmatrix} \right\| \\ &= \sqrt{| f(x_1^{'}) - f(x_1^{''}) |^2 + \cdots + | f(x_n^{'}) - f(x_n^{''}) |^2} \\ &\le \sqrt{L^2 \, | x_1^{'} - x_1^{''} |^2 + L^2 \, | x_2^{'} - x_2^{''} |^2 + \cdots + L^2 \, | x_n^{'} - x_n^{''} |^2} \end{aligned} \qquad (1.9.)$$

S-a trecut la inegalitate utilizând faptul ca f(x) este continua în sens Lipschitz.

Se deduce mai departe ca

$$\| \mathbf{y}' - \mathbf{y}'' \| = L \left\| \begin{matrix} x_1' - x_1'' \\ x_2' - x_2'' \\ \vdots \\ x_n' - x_n'' \end{matrix} \right\| = L \left\| \mathbf{x}_1' - \mathbf{x}_1'' \right\| \tag{1.10.}$$

Introducând în relatia de mai sus termenul în y se obtine:

$$\left\| \mathbf{h}(\mathbf{x}', t) - \mathbf{h}(\mathbf{x}'', t) \right\| \le \left(1 + L \left\| \hat{\mathbf{A}} \right\| \right) \left\| \mathbf{x}_1' - \mathbf{x}_1'' \right\| = \hat{L} \left\| \mathbf{x}_1' - \mathbf{x}_1'' \right\| \tag{1.11.}$$

în care putem identifica un nou L de forma $\hat{L} = 1 + L\left\|\hat{\mathbf{A}}\right\|$.

Deducem ca exista o constanta pentru care se verifica inegalitatea pentru cazul în care functia este h(x,t), pentru orice x’ si x” apartinând lui R. Deci h este continua în sens Lipschitz in raport cu variabila x.
Mai mult, deoarece h(x,t) este continua în raport cu x si este continua pentru toti t (datorita ipotezei 2) exista un M(x_0,t_0,a), astfel încât:

$$\left\| \mathbf{h}(\mathbf{x_0}, \mathrm{t}) \right\| \le \mathrm{M}_{\mathbf{x}_0, t_0, a} \forall \ \ \mathrm{t} \in [t_{0,} t_0 + a] \tag{1.12.}$$

Atunci, pentru orice x în asa fel încât $\left\| \mathbf{x} - \mathbf{x_0} \right\| \le$ b, $\quad$ t $\in [t_{0,} t_0 + a]$,

$$\left\| \mathbf{h}(\mathbf{x}, \mathrm{t}) \right\| \le \mathrm{M}_{\mathbf{x}_0, t_0, a} + \hat{L} \bullet b \tag{1.13.}$$

Rezulta conform teoremei Picard-Lindelof ca exista o solutie unica pentru o durata de timp de

$$\min(\mathrm{a}, \frac{\mathrm{b}}{\mathrm{M}_{\mathbf{x}_0, t_0, a} + \hat{L} \bullet b}) \tag{1.14.}$$

Alegând b destul de mare si $a > \frac{1}{\hat{L}}$, se poate vedea ca exista o solutie pentru aproximativ un timp de $\frac{1}{\hat{L}}$ secunde în care $\hat{L}$ este independent de x_0, t_0, a si b.
Se utilizeaza acelasi procedeu pentru a arata ca exista o solutie unica si pentru

urmatoarele $\frac{1}{\hat{L}}$ secunde, s.a.m.d. Concluzia este ca exista o solutie unica în timp.

Marginirea solutiei

Teorema 2

Pentru ipoteza în care se satisfac urmatoarele conditii $|x_{ij}(0)|\le 1$, $|u_{ij}(t)|\le 1$, $|z_{ij}(t)|\le z_{max}$, solutia $x_{ij}(t)$ a CNN-ului standard cu operatori sinaptici liniari si fara memorie este marginita uniform în sensul ca exista o constanta x_{max} astfel încât $|x_{ij}(t)| \le x_{max}$ $\forall t\ge 0$, $1\le i\le M$, $1\le j\le N$, în care x_{max} este dat de relatia:

$$x_{max} = 1 + z_{max} + \max_{\substack{0\le i\le M \\ 0\le j\le N}} \Big[\sum_{C(k,l)\in Sr(i,j)} (|A(i,j;k,l)| + |B(i,j;k,l)|) \Big] \tag{1.15.}$$

Demonstratie:

Ecuatia ce descrie un CNN standard este:

$$\dot{x}_{ij} = -x_{ij} + \underbrace{\underbrace{\sum_{C(k,l)\in Sr(i,j)} A(i,j;k,l)\,y_{kl}}_{a_{ij}(t)} + \underbrace{\sum_{C(k,l)\in Sr(i,j)} B(i,j;k,l)\,u_{kl}}_{b_{ij}(u(t))} + z_{ij}}_{f(t)} \tag{1.16.}$$

Solutia acestei ecuatii este data mai jos:

$$x(t) = x_{ij}(0)\,e^{-t} + \int_0^t e^{-(t-\tau)}[a_{ij}(\tau) + b_{ij}(u(\tau)) + z_{ij}(\tau)]\,d\tau \tag{1.17.}$$

Aplicând inegalitatea triunghiului, se obtine:

$$\begin{aligned} x(t) &\le |x_{ij}(0)\,e^{-t}| + \int_0^t e^{-(t-\tau)}[|a_{ij}(\tau)| + |b_{ij}(u(\tau))| + |z_{ij}(\tau)|]\,d\tau \\ &\le |x_{ij}(0)\,e^{-t}| + (a_{max} + b_{max} + z_{max})\int_0^t e^{-(t-\tau)}\,d\tau \end{aligned} \tag{1.18.}$$

în care:

$$a_{max} = \max_{t \geq 0} |a_{ij}(t)| \leq \sum_{C(k,\ l) \in Sr(i,\ j)} |A(i,\ j; k, l)| \max_{t \geq 0} y_{kl}(t) \tag{1.19.}$$

$$b_{max} = \max_{u} |b_{ij}(u)| \leq \sum_{C(k,\ l) \in Sr(i,\ j)} |B(i,\ j; k, l)| \max_{u} u_{kl}(t)$$

Se tine seama de inegalitatea:

$$\int_0^t e^{-(t-\tau)} d\tau = 1 - e^{-t} < 1,\ t \geq 0 \tag{1.20.}$$

cu $|u_{kl}(t)| \leq 1$, $|x_{ij}(0)| \leq 1$.

Se deduce deci ca $x_{ij}(t)$ este marginit de valoarea:

$$\begin{aligned} |x_{ij}(t)| &\leq |x_{ij}(0)\, e^{-t}| + a_{max} + b_{max} + z_{max} \leq 1 + z_{max} + \\ &\sum_{C(k,l) \in Sr(i,\ j)} |A(i,\ j; k, l)| + \sum_{C(k,l) \in Sr(i,\ j)} |B(i,\ j; k, l)| \\ &\leq 1 + z_{max} + \max_{\substack{0 \leq i \leq M \\ 0 \leq j \leq N}} \left[\sum_{C(k,l) \in Sr(i,\ j)} (|A(i,\ j; kl)| + |B(i,\ j; kl)|) \right] = x_{max} \end{aligned} \tag{1.21.}$$

care este independenta de perechea (i,j).

Observatie
Pentru CNN-uri invariante în spatiu $|x_{ij}(t)|$ este marginit de valoarea:

$$\hat{x}_{max} = 1 + z_{max} + \sum_{1 \leq k \leq M} \sum_{1 \leq l \leq N} |A_{kl}| + |B_{kl}| \tag{1.22.}$$

Teorema 2 impune o tensiune de alimentare minima pentru oricare implementare de circuit a CNN-urilor si este fundamantala pentru proiectarea unui circuit integrat ce implementeaza un CNN.

În continuare se prezinta intuitiv influenta template-urilor A si B asupra comportarii CNN-ului, precum si o serie de cazuri particulare pentru forma acestora. Se disting astfel mai multe clase de CNN-uri, ce se pot analiza separat.

Definitia 6: ***Ponderi excitatorii si inhibitorii***

Se spune ca o pondere (element al template-ului A) [12] a_{kl} este excitatoare (respectiv inhibitoare) daca este pozitiva, respectiv negativa. Aceasta se justifica prin faptul ca cele excitatoare determina cresterea functiei $h_{ij}(x_{ij},w_{ij})$ si deci a vitezei de crestere a lui $x_{ij}(t)$, pe când cele inhibitoare determina scaderea acestei viteze de crestere a starii.
Cazul cel mai general pentru CNN-ul standard, în care toate elementele template-urilor *pot fi* nenule este prezentat mai jos. Starea celulei din pozitia (i,j) este determinata de iesirile celulelor vecine si de intrarile corespunzatoare celulelor vecine (pixeli din imaginea de pe intrare cu indecsi corespunzatori celulelor vecine).
Rezulta astfel clasa de CNN standard notata cu C(A,B,z).

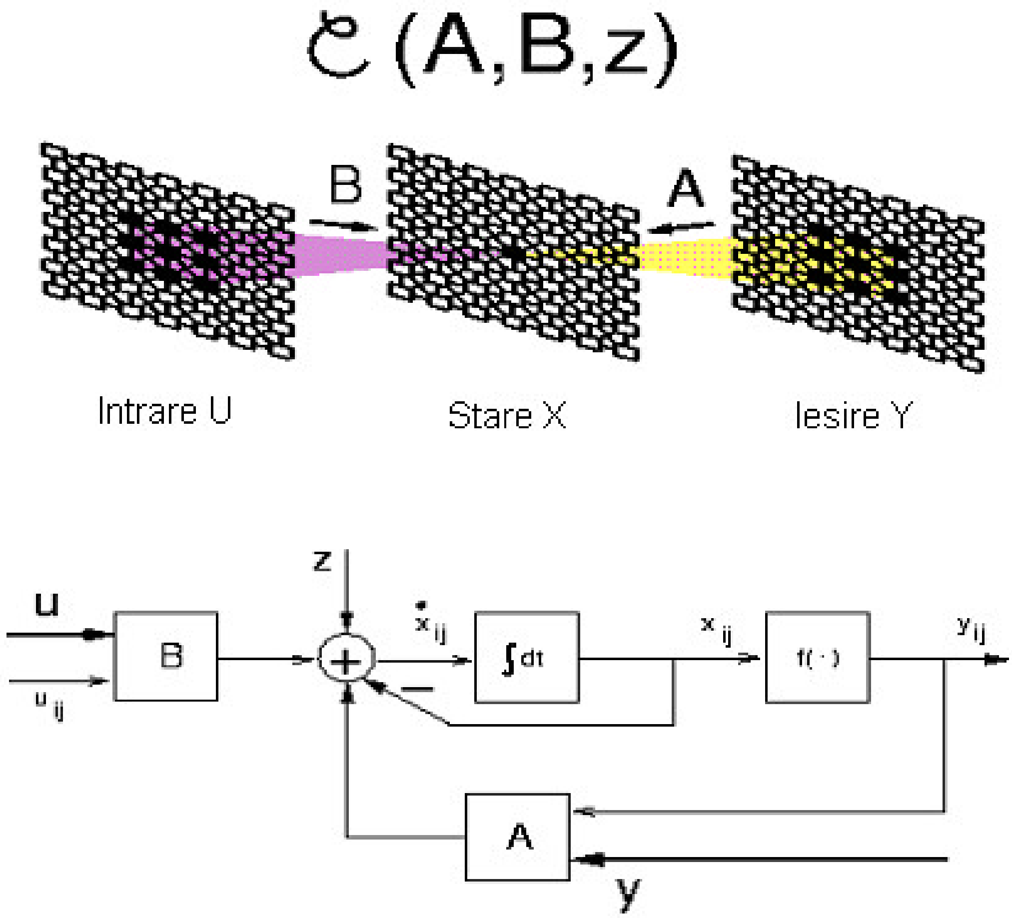


Fig. 5: a) Reprezentarea intuitiva a fluxului de date în CNN
b) Reprezentarea sistemica a unei celule din CNN

În Fig. 5a) s-a schitat fluxul de date în CNN, iar în Fig. 5b) s-a reprezentat folosind elemente de sistem o celula din pozitia (i,j) interconectata cu vecinii.

Definitia 7: ***Clasa cu reactie zero C(0,B,z)***

Un CNN apartine clasei C(0,B,z) [12] daca si numai daca celula este influentata numai de intrarile corespunzatoare celulelor vecine.

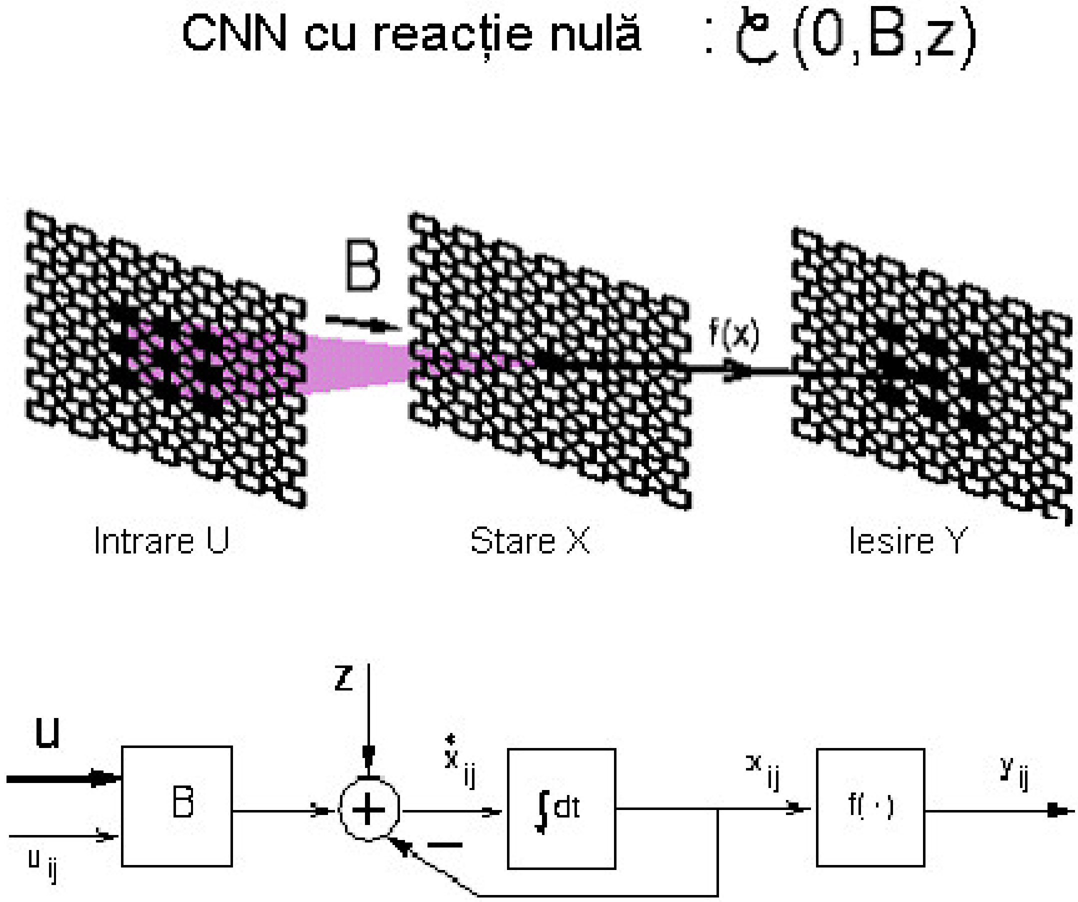


Fig. 6: a) Reprezentarea intuitiva a fluxului de date în CNN-ul fara reactie
b) Reprezentarea sistemica a CNN-ului fara reactie

*Definitia 8***: *Clasa CNN-urilor autonome C(A, 0, z)***

Un CNN apartine clasei C(A, 0, z) [12] daca si numai daca toate elementele apartinând template-ului de intrare sunt zero.

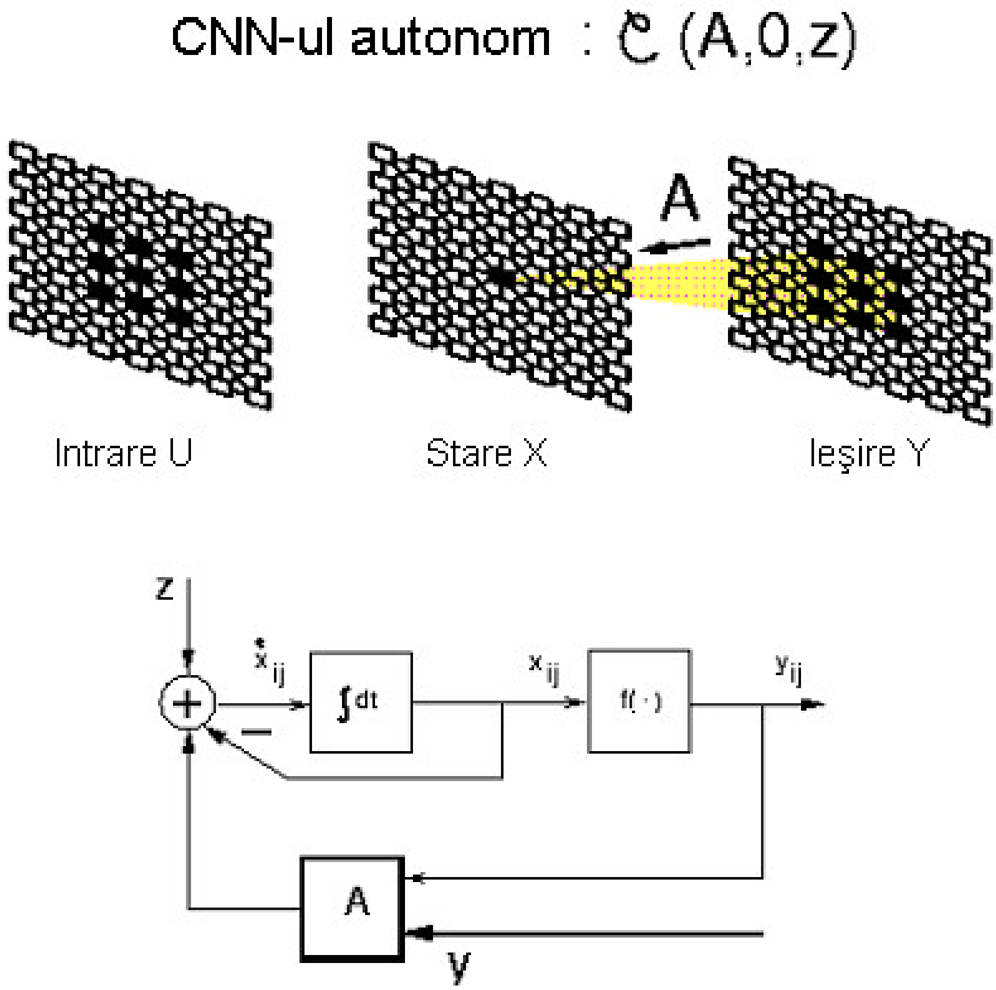
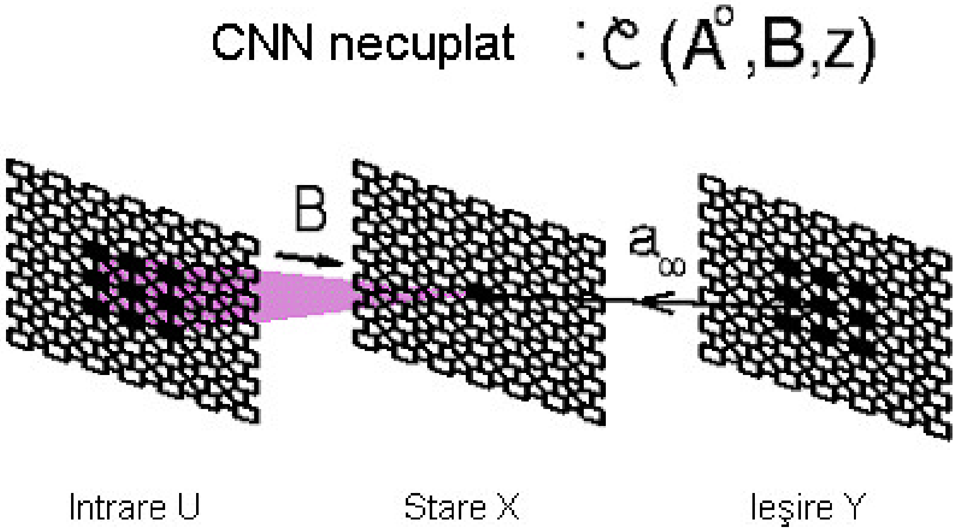


Fig. 7: a) Reprezentarea intuitiva a fluxului de date în CNN-ul autonom
b) Reprezentarea sistemica a CNN-ului autonom

Definitia 9: ***Clasa CNN cu celule necuplate C(A^0, B, z)***

Un CNN apartine clasei cu celule necuplate C(A^0, B, z) daca si numai daca a_{ij}=0, pentru oricare i si j cu exceptia lui i=j.



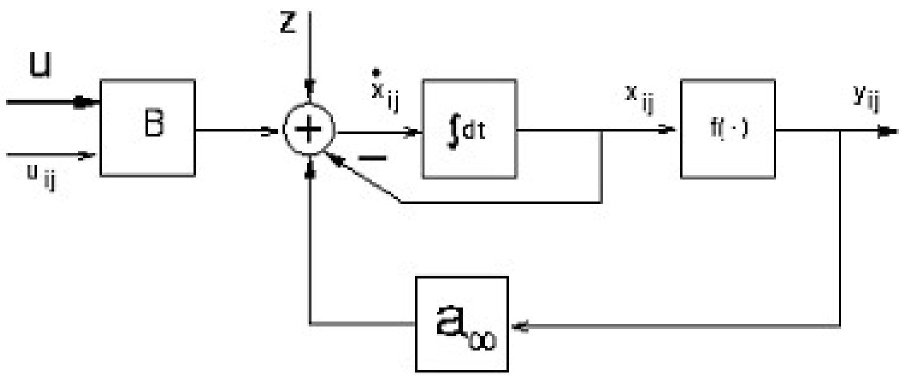


Fig. 8: a) Reprezentarea intuitiva a fluxului de date în CNN-ul necuplat
b) Reprezentarea sistemica a CNN-ului necuplat

Capitolul 2.: Functionarea CNN-ului cu celule de ordinul I ca filtru liniar spatial

O clasa de aplicatii ale CNN-ului in prelucrarea imaginilor consta in utilizarea retelei in zona liniar centrala a celulelor [28,29].

Functionarea CNN în regiunea central liniara a celulelor componente

Retelele neuronale celulare pot fi privite ca filtre spatiale liniare. În cazul în care în urma evolutiei valorilor variabilelor de stare asociate celulelor nu s-a intrat înca în neliniaritate avem $y_{i,j}=x_{i,j}$ pentru cazul 2D, care este cel mai general[1].
Se ia în discutie urmatoarea forma a sistemului de ecuatii ce descriu CNN-ul standard omogen (adica template-urile A, B si pragul z sunt aceleasi pentru orice pereche (i,j)):

$$\dot{x}_{ij(t)} = -x_{ij(t)} + \sum_{C(k,l)\in Sr(i,j)} A(k,l)x_{kl} + \sum_{C(k,l)\in Sr(i,j)} B(k,l)u_{kl} + z \tag{2.1.}$$

Se fac urmatoarele notatii:

$$a(n_1,n_2) = \begin{cases} A_{0,0} - 1, & (n_1,n_2)=(0,0) \\ A_{-n_1,-n_2}, & -n_1,-n_2 \in N \\ 0 \quad , & \text{în} \quad rest \end{cases}$$

(2.2.)

$$b(n_1,n_2) = \begin{cases} B_{-n_1,-n_2}, & -n_1,-n_2 \in N \\ 0 \quad , & \text{în} \quad rest \end{cases}$$

Cu aceste notatii se poate deduce ecuatia care guverneaza dinamica CNN-ului de ordinul I:

$$\dot{x}_t(n_1,n_2) = a(n_1,n_2) * x_t(n_1,n_2) + b(n_1,n_2) * u(n_1,n_2) + z \tag{2.3.}$$

Abordarea problemei în domeniul frecventelor spatiale

Solutia ecuatiei (2.3) se poate obtine utilizând Transformata Fourier Discreta (TFD) sau Transformata Cosinus Discreta (TCD) în functie de conditiile la limita cu care se lucreaza:

$$F(w_1,w_2) = \sum_{n_1}\sum_{n_2} f(n_1,n_2)e^{-jw_1n_1}e^{-jw_2n_2} \tag{2.4.}$$

Câteva proprietati interesante pentru problematica de fata ale TFD sunt:

---

[1] Cazul 1D se poate obtine eliminând un indice al variabilelor de tip semnal

- daca $f(n_1, n_2)$ este simetrica si reala, atunci si $F(\omega_1, \omega_2)$ este simetrica si reala;
- f(0,0) deplaseaza suprafata $F(\omega_1, \omega_2)$ în sus si în jos;
- $TFD\{f(n_1, n_2)*g(n_1, n_2)\}=F(\omega_1, \omega_2)\, G(\omega_1, \omega_2)$;

În cazul în care se considera o retea discreta infinita, se poate scrie:

$$\dot{X}_t(w_1,w_2) = A(w_1,w_2)X_t(w_1,w_2) + B(w_1,w_2)U(w_1,w_2) + zd(w_1,w_2) \quad (2.5.)$$

În cele ce urmeaza vom considera un CNN cu dimensiunile N_1xN_2 si conditii la limita periodice (caz în care se utilizeaza TFD) respectiv flux-zero (pentru care se utilizeaza TCD).

Toate consideratiile care urmeaza se refera la CNN-ul stabil. Pentru domeniul frecventa discretizat,

$$\dot{X}_t(k_1,k_2) = A(\frac{2pk_1}{N_1},\frac{2pk_2}{N_2})X_t(k_1,k_2) + B(\frac{2pk_1}{N_1},\frac{2pk_2}{N_2})U(k_1,k_2) + zd(k_1,k_2) \quad (2.6.)$$

cu

$$w_1 = \frac{2pk_1}{N_1} \quad si \quad w_2 = \frac{2pk_2}{N_2} \quad (2.7.)$$

Pentru cazul în care se lucreaza cu TCD, se obtine un set de relatii asemanatoare:

$$\dot{X}_t(k_1,k_2) = A(\frac{pk_1}{N_1},\frac{pk_2}{N_2})X_t(k_1,k_2) + B(\frac{pk_1}{N_1},\frac{pk_2}{N_2})U(k_1,k_2) + zd(k_1,k_2) \quad (2.8.)$$

cu:

$$w_1 = \frac{pk_1}{N_1} \quad si \quad w_2 = \frac{pk_2}{N_2} \quad (2.9.)$$

Solutia în timp

Vom scrie solutia în domeniul timp cu frecventa neexplicitata (variabila analogica) pentru a nu opta in acest moment in mod necesar pentru una din TFD sau TCD:

$$\begin{aligned} &\textit{Când } A(v_1,v_2) \neq 0: \\ &X_t(v_1,v_2) = e^{A(v_1,v_2)t} X_0(v_1,v_2) + \\ &\frac{1}{A(v_1,v_2)}\left[e^{A(v_1,v_2)t} - 1\right]B(v_1,v_2)U(v_1,v_2) \end{aligned} \quad (2.10.)$$

$$\begin{aligned} &\textit{Când } A(v_1,v_2) = 0: \\ &X_t(v_1,v_2) = X_0(v_1,v_2) + tB(v_1,v_2)U(v_1,v_2) \end{aligned} \quad (2.11.)$$

Pentru a face notatiile mai simple si mai intuitive, s-a abandonat punerea în evidenta a termenului care tine de sursele de curent constant. Acest termen se poate include în termenul excitatie.

Functionarea CNN-ului stabil în regiunea central liniara

Filtrarea liniar spatiala a semnalului excitatie

În cazul în care pentru toate perechile (ω_1,ω_2) este valabila relatia[2]:

$$\mathrm{Re}(A(w_1,w_2)) < 0 \tag{2.12.}$$

atunci sistemul liniar central este stabil, iar exponentialele corespunzatoare fiecarui mod vor descreste în timp.
Punctul de echilibru stabil, care este independent de conditia initiala se poate scrie:

$$X_\infty(w_1,w_2) = H(w_1,w_2)U(w_1,w_2) \tag{2.13.}$$

în care functia de transfer H(ω_1, ω_2) este:

$$H(w_1,w_2) = \frac{-1}{A(w_1,w_2)}B(w_1,w_2) \tag{2.14.}$$

Se recunoaste în relatia (2.14) o functie de transfer a unui filtru IIR.
Cel mai simplu filtru care se poate imagina folosind CNN-ul este cel în care se utilizeaza pentru filtrare doar template-ul B.
Daca exista doar element central în template-ul A, si acesta este o constanta γ < 1, atunci functia de transfer pentru aceasta situatie are forma:

$$H(w_1,w_2) = \frac{-1}{g-1}B(w_1,w_2) \tag{2.15.}$$

Când se alege si γ=0, se obtine un filtru FIR corespunzator template-ului B.

[2] Se recunoaste în relatia respectiva valorile proprii temporale ale sistemului si se foloseste definitia stabilitatii dupa Lyapunov

Filtrarea liniara spatiala variabila în timp

În cazul în care CNN-ul este excitat prin conditii initiale, evolutia starii este data de ecuatia:

$$X_t(w_1,w_2) = H(w_1,w_2)X_0(w_1,w_2) \quad (2.16.)$$

unde:

$$H(w_1,w_2) = e^{A(w_1,w_2)t} \quad (2.17.)$$

CNN-ul se comporta ca un filtru spatial variabil in timp.

Functionarea CNN-ului instabil în regiunea central liniara

Sistemul liniar central este instabil, daca exista cel putin o pereche (ω_1, ω_2) pentru care:

$$\mathrm{Re}(A(w_1,w_2)) > 0 \quad (2.18.)$$

Dinamica CNN-ului autonom instabil

În esenta, diferenta fata de CNN-ul stabil consta în aceea ca, datorita instabilitatii, unele moduri spatiale pot creste în timp.
Se obtin aceleasi relatii ca în cazul filtrarii liniare spatiale în timp cu mentiunea ca în acest caz sistemul este instabil. Acest caz va fi îndelung discutat pentru CNN-ul format din celule de ordin superior si template-uri de ordin superior (cu r>1), prezentându-se o serie de metode de proiectare pentru aceste cazuri. Cazul CNN-ului cu celule de ordinul I va fi privit ca o particularizare a acestor cazuri, în acest capitol realizându-se doar o analiza a CNN-ului standard omogen. În acest caz avem de a face cu o “filtrare liniara functionala” variabila în timp dar numai pâna în momentul în care prima celula intra în neliniaritate [30], adica variabila de stare “trece” printr-o functie neliniara de tip amplificare cu saturatie.

Capitolul 3.: Functionarea neliniara a CNN-ului standard. Strategii pentru proiectarea de template-uri. Proiectare robusta

Scopul proiectarii unui template este de a obtine o prelucrare impusa a imaginilor. De exemplu, extragere de contururi, segmentare, etc.
Exista trei metode de proiectare a template-urilor [31-50]:

- metoda intuitiva;
- metoda învatarii;
- deducerea directa de template-uri.

Metoda intuitiva cere judecata intuitiva a proiectantului. Necesita experienta în procesare de imagini si dinamica retelelor.

Metoda învatarii – se bazeaza pe:

- perechi intrare – iesire;
- în unele cazuri nu exista template-ul pentru o problema data;
- câteva metode de învatare propuse au fost testate sa fie eficiente pentru câteva template-uri binare intrare – iesire. Atâta timp cât cea mai mare parte dintre aceste template-uri pot fi deduse direct, valoarea acestei metode poate fi pusa la îndoiala. În aceste cazuri acolo unde nu este o iesire dorita explicita, proiectarea directa de template-uri este imposibila. Segmentarea texturilor este un exemplu bun în acest sens.

Deducerea directa de template-uri – poate fi aplicata acolo unde procesarea dorita poate fi descrisa la nivel local în mod exact. Metodele de proiectare depind de clasa de CNN particulara (C(A,B,z), C(0,B,z), etc). În timp ce cu primele doua metode rezulta câteva template-uri valide, aceasta metoda permite alegerea celei mai robuste solutii dintr-o multime de solutii. Mai mult, aceasta metoda este "ieftina" computational.

Template-uri CNN necuplate

Forma generala a unui astfel de template este:

$$A = \begin{bmatrix} 0 & 0 & 0 \\ 0 & a_{00} & 0 \\ 0 & 0 & 0 \end{bmatrix} \quad B = \begin{bmatrix} b_{-1-1} & b_{-10} & b_{-11} \\ b_{0-1} & b_{00} & b_{01} \\ b_{1-1} & b_{10} & b_{11} \end{bmatrix} \quad z = i \qquad (3.1.)$$

Pentru acest tip de template, sistemul lucreaza ca o retea de celule independente. Prin urmare, studiul comportarii dinamicii unei singure celule este suficient pentru a trage concluziile asupra comportarii întregii retele.
O celula primeste noua intrari fixe de la celulele vecine. Are o conditie initiala x(0) care poate fi considerata a zecea intrare. Prin urmare, se implementeaza o functie de tipul 10 intrari si o iesire.

Ecuatia de stare a unei singure celule are expresia:

$$\dot{x}(t) = -x(t) + a_{00}\, y(t) + s$$
$$s = \sum_{C(k,l)\in N_r(i,j)} B_{ij,kl} u_{kl} + z, \quad y = f(x) \tag{3.2.}$$

Parametrul s este evident o constanta, dat fiind faptul ca depinde de template si de intrarea în CNN (nu se ia în calcul intrarea variabila în timp).
Forma functiei f(x) este data în Fig. 9:

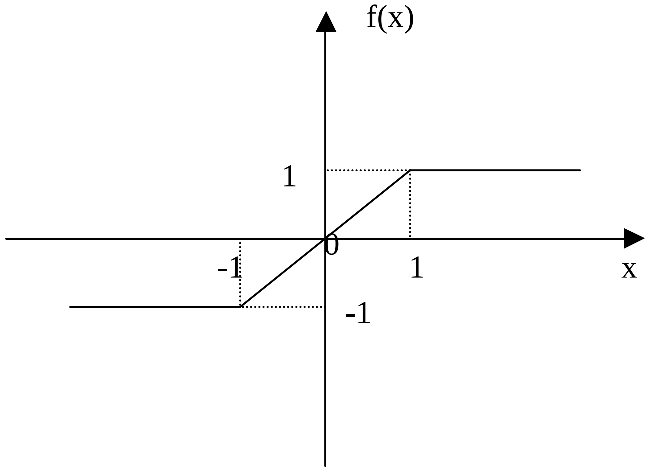


Fig. 9: Forma neliniaritatii

Reprezentarea cu elemente de sistem este data mai jos:

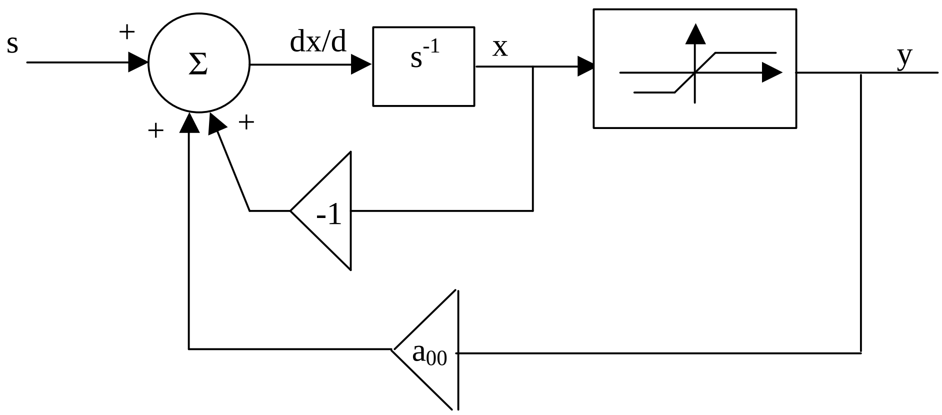


Fig. 10: Reprezentarea cu elemente de sistem a CNN-ului necuplat

Reguli de proiectare în contextul CNN cu celule necuplate

Regula 1: $a_{00}=0$, rezulta $y_\infty=f(s)$

Observatii:
- iesirea nu depinde de x(0);
- regimul tranzitoriu este exponential.

Regula 2: $a_{00}=1$, rezulta ca CNN-ul este un integrator atâta timp cât modulul lui x este subunitar

2a) Daca s este diferit de zero, rezulta ca $y_\infty=\text{sign}(s)$;

2b) Daca s este nul, $y_\infty=x(0)$.

Observatii:
- daca s este diferit de zero atunci y_∞ nu depinde de x(0) si este binar;
- iesirea se satureaza în mai putin de $2s\tau$, în care τ este constanta de timp a celulei;
- în practica trebuie sa evitam ca s sa fie nul deoarece aceasta solutie nu este robusta în cazul unei realizari analogice.

Regula 3: $a_{00}>1$, iesirea este tot timpul binara, datorita buclei de reactie pozitiva. Iesirea depinde de starea x(0) si de contributia lui s. Sunt trei cazuri de importanta practica:

3a) x(0)=0, rezulta y(0)=0. Iesirea depinde de semnul lui s în felul urmator: $y_\infty=\text{sign}(s)$;

3b) x(0)=1, rezulta y(0)=1. Iesirea finala:
- ramâne 1, daca $a_{00}+s>1$;
- se schimba în –1 daca $a_{00}+s<1$.

3c) x(0)=-1, rezulta y(0)=-1. Iesirea finala:
- ramâne –1 daca $-a_{00}+s<-1$;
- se schimba în 1 daca $-a_{00}+s>-1$.

3d) $x(0)=x_0$, modulul lui x_0 subunitar, rezulta ca $y(0)=x_0$,

$y_\infty=\text{sign}(dx/dt(x=x_0))=\text{sign}(a_{00}x_0-x_0+s)=\text{sign}((a_{00}-1)x_0+s)$. Aceasta regula este generalizarea celor doua de mai sus.

Observatie: $a_{00}+s=1$ si $-a_{00}+s=-1$ trebuie evitate tinând seama de erorile implicate în realizarea analogica.

În deducerea template-urilor exista o prima etapa în care se deduce forma acestora (adica parametrii independenti si pozitiile lor în template) si o a doua, în care se deduc în urma rezolvarii unui sistem de inecuatii valorile parametrilor template-urilor.

Template-uri pentru prelucrari binare

În aceasta categorie intra prelucrarea imaginilor formate din pixeli albi si negri. Fiind în cadrul clasei de CNN-uri necuplate, se vor folosi pentru prezentarea urmatoarelor clase de prelucrari de imagine regulile prezentate mai sus.
Au fost grupate principalele template-uri în trei clase:

Clasa 1: Extragere de forme[1].

Acest grup de template-uri extrage forme bine definite. Când se descrie o problema, se determina o masca 3x3 si un numar întreg. Masca binara contine pixeli negrii, pixeli albi si pixeli care nu conteaza.

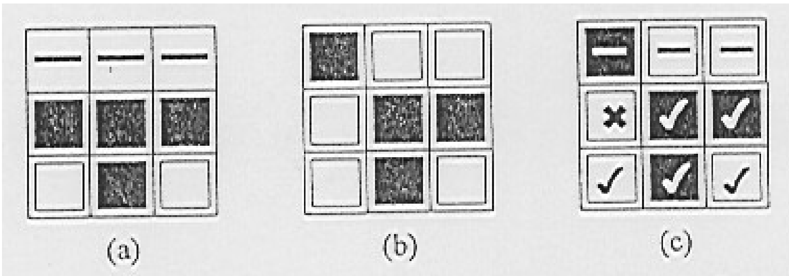


Fig. 11: Reguli locale pentru extragerea de trasaturi
a) masca; b) imagine de test; c) potrivirea mastii pe imaginea de test

Numarul întreg reprezinta numarul minim de pozitii ale mastii (Fig. 11a) care ar trebui sa se potriveasca pe o regiune din imagine (Fig. 11b) pentru a extrage (schimba în negru) pixelul central din aceasta. Cu alte cuvinte, iesirea este neagra în acele pozitii unde sunt cel putin "prag" pixeli care se potrivesc cu forma dorita. În Fig. 11c se prezinta modul în care masca se potriveste pe o regiune de 3x3 pixeli din imaginea-sursa.

Metoda de proiectare a template-urilor pentru procesari binare pe intrare binara:

Metoda este ilustrata în figura:

Determinare structura template B → Generare sistem inecuatii → Gasire solutie sistem → Alegerea celui mai robust template

Fig. 12: Metoda de proiectare a template-urilor pentru procesari binare pe intrare binara

Determinarea formei template-ului B este cea mai importanta operatie. Prin aceasta se realizeaza reducerea spatiului parametrilor care are initial dimensiunea 11. Uzual dimensiunea acestui spatiu al parametrilor template-ului se reduce la 3, 4.

[1] Forme este traducerea englezescului *pattern*

Generarea sistemului de inecuatii se face tinând cont de procesarea ce trebuie facuta si de regulile mai sus mentionate.
Rezolvarea sistemului de inecuatii presupune intersectia regiunilor hiperspatiului care satisfac toate inecuatiile sistemului. Daca nu exista nici o regiune comuna, atunci procesarea nu se poate face cu ajutorul template-urilor liniare.
Alegerea celui mai robust template. Cel mai robust template este cel care corespunde regiunii din mijlocul portiunii de hiperspatiu determinate.

Exemplu: Determinarea unui template pentru clasa 1 (extragere de forme)
În acest exemplu se urmareste extragerea acelor pixeli din imaginea sursa pentru care 5 pozitii în masca se potrivesc cu „petecul" din imaginea de la intrare. Se lucreaza cu situatia prezentata în Fig. 11.

Sunt valabile urmatoarele propozitii:

- iesirile trebuie sa fie binare. Rezulta ca $a_{00}>1$, conform cu Regula 3.
- se presupune ca x(0)=0. Rezulta ca y_∞=sign(s) conform cu Regula 3.
- în template-ul B, pozitiile care nu conteaza vor fi nule.
- pozitiile cu negru trebuie sa se supuna toate aceleiasi reguli, de unde rezulta ca vor fi caracterizate de o variabila independenta. Se noteaza cu b. Cele cu alb joaca acelasi rol între ele si opus pozitiilor cu negru de unde rezulta ca vor fi caracterizate de variabila –b.

Forma template-ul ce rezulta este descris mai jos:

$$A = \begin{bmatrix} 0 & 0 & 0 \\ 0 & a & 0 \\ 0 & 0 & 0 \end{bmatrix} \qquad B = \begin{bmatrix} 0 & 0 & 0 \\ b & b & b \\ -b & b & -b \end{bmatrix} \qquad z = i \qquad (3.3.)$$

Generarea sistemului de inecuatii impune analiza tuturor combinatiilor posibile ale mastii si aplicarea regulilor, în acest caz Regula 3

Numarul de pixeli care se potrivesc cu masca	Iesire dorita	Inecuatie
6	negru (+1)	6b+i>0
5	negru (+1)	4b+i>0
4	alb(-1)	2b+i<0
3	alb(-1)	i<0
2	alb(-1)	-2b+i<0
1	alb(-1)	-4b+i<0
0	alb(-1)	-6b+i<0

Combinatiile se genereaza cu ajutorul unui program (numar de posibilitati: 2 la puterea a noua). Se vede din tabel ca pentru un numar de "potriviri" de pixeli din

petecul din imaginea sursa si masca mai mare sau egal cu 5, iesirea dorita este negru (1). Pentru celelalte situatii, pixelul de la iesire se doreste a fi alb (-1).
Se arata usor ca, cu cât valorile din template sunt mai mari, cu atât regimul tranzitoriu se stinge mai repede.

Rezolvarea grafica a sistemului de inecuatii este figurata mai jos, în planul parametrilor a si b

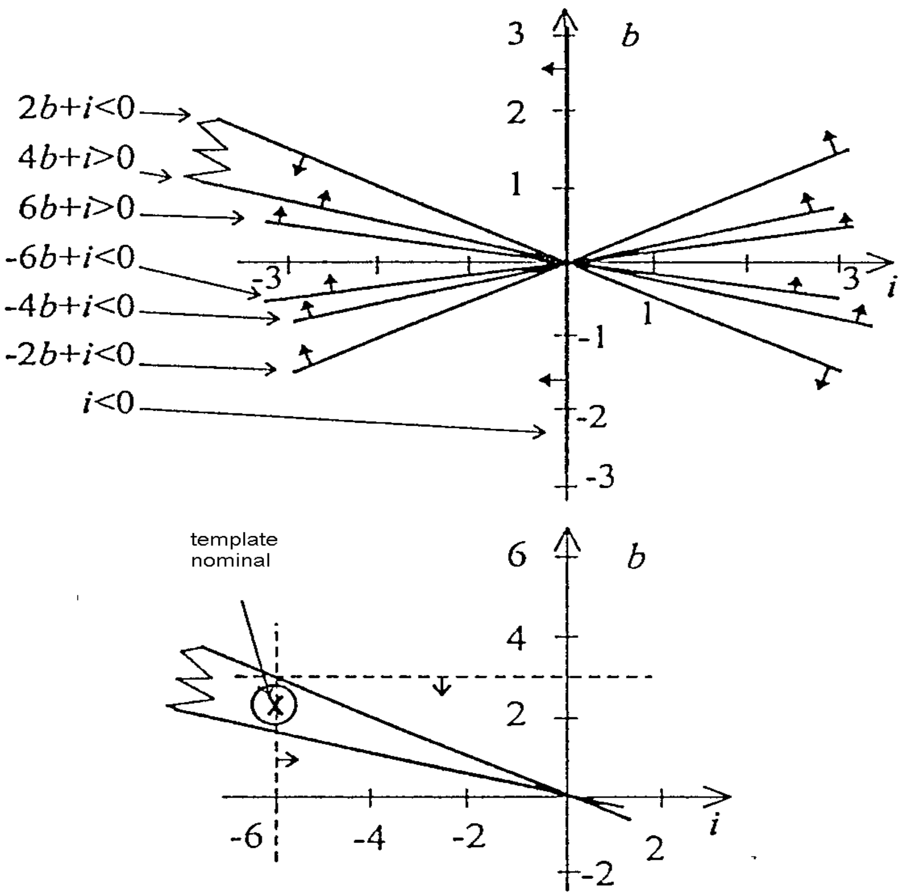


Fig. 13: Rezolvarea grafica a sistemului de inecuatii

Realizarea analogica a CNN-ului limiteaza însa valoarea absoluta maxima a valorilor din template. Aceasta limiteaza subspatiul infinit la unul finit. Cel mai bun template este tinând cont atât de viteza de procesare cât si de **robustete** urmatorul:

$$A = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 0 \end{bmatrix} \quad B = \begin{bmatrix} 0 & 0 & 0 \\ 2.2 & 2.2 & 2.2 \\ -2.2 & 2.2 & -2.2 \end{bmatrix} \quad z = -6 \qquad (3.4.)$$

Observatii:

- o proprietate speciala a acestei clase de template-uri este aceea ca template-ul B contine doar un parametru independent (în cazul exemplului prezentat, acesta este b).
- din punctul de vedere al robustetii, daca termenul de tip prag (z) ar fi fost mai mare (-5 în loc de –6), "fâsia" cu spatiul în care ar fi putut exista solutia ar fi fost mai îngusta. Acest lucru ar fi facut mai dificila plasarea unui cerc în interiorul zonei corespunzatoare solutiilor sistemului (vezi figura precedenta). Acest cerc da plaja în care pot varia (tehnologic sau prin alt mecanism) parametrii determinati pentru ca template-ul sa realizeze procesarea dorita.

Template-urile din biblioteca de template-uri care fac parte din aceasta clasa sunt: Erodare[2], Dilatare, StergeFigura, si altele. Prelucrarile de imagini realizate se deduc usor din numele template-urilor respective.

Clasa 2: Modificare conditionala de pixel

Daca la clasa anterioara se construia o imagine la iesire bazata pe o imagine de la intrare, imaginea de la iesire putând fi însa mult diferita fata de cea de la intrare, aici imaginea de pe iesire este apropiata de imaginea sursa, folosita ca intrare în CNN, doar câtiva pixeli fiind schimbati.

În cazul unui template FUNCTIONAL ASIMETRIC (z este nenul) ori se schimba câtiva pixeli negrii în pixeli albi, ori invers. Niciodata nu se schimba si cei albi în negrii si cei negrii în albi.

În cazul unui template FUNCTIONAL SIMETRIC (z=0), daca se satisface o conditie[3], pixelii negrii se schimba în pixeli albi iar daca opusul conditiei este satisfacuta, pixelii albi devin pixeli negrii. Pixelii din regiunile în care nu este satisfacuta conditia nu se schimba.

Aici se dau ca date de intrare: o masca binara, un prag (numar întreg) si o regula prin care fie pixelii negrii se schimba în pixeli albi, fie invers.

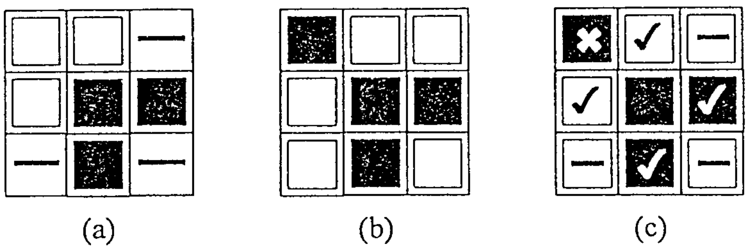


Fig. 14: Reguli locale
a) masca; b) imagine de test; c) potrivirea mastii pe imaginea de test

Exemplu:

Se doreste determinarea unui template care sa realizeze urmatoarea procesare de imagine: sa se schimbe acei pixeli negrii ale caror vecinatati se potrivesc cu masca data în 5 pozitii si lasa ceilalti pixeli (albi sau negrii) neschimbati.

[2] Numele îsi au originea din engleza: EROSION, DILATION, FIGDEL
[3] Conditia este una de tipul “un numar de pixeli se potrivesc peste o masca”

Determinarea formei template-ului se face dupa regulile de mai jos:

- iesirile trebuie sa fie binare. Rezulta ca $a_{00}>1$, conform cu Regula 3.
- $x(0)=x_0$, rezulta ca iesirea va fi determinata de Regula 3. Pozitiile care nu conteaza vor fi initializate cu 0.
- specific acestei clase este ca pozitia centrala joaca tot timpul alt rol decât cel al altor pozitii. Prin urmare, centrul template-ului B primeste un parametru independent (b).
- pozitiile negre (fara pixelul din mijloc) joaca dupa aceleasi reguli. Rezulta ca li se atribuie parametrul c. Rolul pozitiilor albe este exact inversul rolului celor negre.

$$A = \begin{bmatrix} 0 & 0 & 0 \\ 0 & a & 0 \\ 0 & 0 & 0 \end{bmatrix} \qquad B = \begin{bmatrix} -c & -c & 0 \\ -c & b & c \\ 0 & c & 0 \end{bmatrix} \qquad z = i \qquad (3.5.)$$

Generarea sistemului de inecuatii este prezentata în continuare.
Trebuiesc analizate toate combinatiile posibile ale pattern-ului de intrare si aplicate regulile, în acest caz Regula 3

Intrare	Numarul pixelilor care se potrivesc	Iesire dorita	Inecuatie
negru(+1)	5	alb(-1)	a+b+5c+i<1
negru(+1)	4	negru(+1)	a+b+3c+i>1
negru(+1)	3	negru(+1)	a+b+c+i>1
negru(+1)	2	negru(+1)	a+b-c+i>1
negru(+1)	1	negru(+1)	a+b-3c+i>1
negru(+1)	0	negru(+1)	a+b-5c+i>1
alb(-1)	5	alb(-1)	-a-b+5c+i<-1
alb(-1)	4	alb(-1)	-a-b+3c+i<-1
alb(-1)	3	alb(-1)	-a-b+c+i<-1
alb(-1)	2	alb(-1)	-a-b-c+i<-1
alb(-1)	1	alb(-1)	-a-b-3c+i<-1
alb(-1)	0	alb(-1)	-a-b-5c+i<-1

Din tabel se vede ca se poate înlocui a+b cu o alta variabila m, dar trebuie sa tinem seama la alegerea ulterioara a lui b ca a este supraunitar.
Template-urile din biblioteca de template-uri care fac parte din aceasta clasa sunt: DetectorDeColturi[4], DetectorDeMuchii, DetectarePunctCentru, etc.

Clasa 3: Functii de doua intrari
Partea care individualizeaza aceasta clasa este ca intrarea si starea initiala a CNN-ului sunt diferite, adica apare o intrare în plus. Numarul total de pixeli care au efect asupra iesirii este 10 (în loc de 9) deoarece în acest caz, pe lânga imaginea introdusa pe intrarea CNN, se mai introduce o imagine si pe stare. La

[4] Numele îsi au originea în denumirile englezesti: CORNER, EDGE, CENTER

acest moment se reaminteste ca este vorba de CNN-uri necuplate. Prin urmare numai pixelul corespunzator pozitiei curente din imaginea introdusa pe stare va interveni în ecuatia de functionare a celulei din acea pozitie. Acest pixel este al zecelea ce intervine (functional) ca intrare în celula curenta.
Imaginea cu care este initializata variabila de stare poate fi folosita (functional) ca o masca de selectie, ce are rol de validare a unei procesari efectuate asupra imaginii introduse pe intrare pe pozitiile selectate de masca.
Se pot folosi pentru imaginile constituite de intrari prelucrarile din clasele 1 si 2. Imaginea rezultata si starea initiala pot fi combinate logic, adica, spre exemplu, acest lucru înseamna ca se poate extrage o forma anume din imaginea de intrare, dar pixelii negri extrasi apar numai în acele pozitii în iesirea finala unde starea initiala a fost negru.

Exemplu:
Se dau doua imagini binare. Iesirea finala va fi negru când ambele intrare si stare initiala au fost negre. Altfel, iesirea este alb.
Determinarea formei template-ului este prezentata în cele ce urmeaza:

- iesirile trebuie sa fie binare. Rezulta ca a_{00}>1, conform cu Regula 3.
- forma template-ului B poate fi dedusa direct din pattern-ul binar, dupa cum s-a aratat în cazurile precedente.
- iesirea finala va fi determinata de Regula 3.

Template-uri care apartin acestei clase în cadrul bibliotecii de template-uri: LOGAND, LOGOR, LOGORN, care realizeaza operatiile logice AND, OR, NOR, etc.

Template-uri pentru CNN-uri cu celule cuplate

Schimbarile în iesirile celulelor afecteaza vecinii si invers, adica mai multe elemente ale template-ului A sunt diferite de zero (clasa C(A,B,z)). Dinamica este descrisa de ecuatia:

$$\dot{x}(t) = -x(t) + \sum_{C(k,l)\in N_r(i,j)} A_{ij,kl}\, y_{kl}(t) + s \quad in\ care\ \ s\ are\ \ forma\ :$$

$$s = \sum_{C(k,l)\in N_r(i,j)} B_{ij,kl}\, u_{kl} + z \qquad iar\ \ y = f(x) \tag{3.6.}$$

Parametrul s este evident o constanta, dat fiind faptul ca depinde de template si de intrarea în CNN (pentru intrare care nu variaza în timp). Forma functiei f(x) este data în Fig. 9:
O particularitate importanta a CNN-urilor cuplate este faptul ca, în anumite cazuri poate avea loc fenomenul de propagare. Problemele care se pun pentru aceasta clasa de CNN-uri sunt:

- cum are loc propagarea?
- cum se poate controla propagarea cu un template proiectat?

În continuare se prezinta câteva notiuni pregatitoare:
În timpul analizei regimului tranzitoriu într-o retea (CNN) la fiecare moment putem separa celulele în *setul activ* si *setul inactiv*.
Celule inactive – o celula este considerata inactiva daca la un moment dat iesirea sa este *stabilizata* la o valoare. O celula inactiva trebuie sa se gaseasca în regiunea de saturatie pentru ca, datorita buclei pozitive, (a_{00}>1) celula nu se poate afla în punct de echilibru stabil în regiunea liniara.
Analizând caracteristica celulei, se poate spune ca o celula este stabila în regiunea de saturatie pozitiva daca:

$$\sum_{C(k,l)\in N_r(i,j)} A_{ij,kl}\, y_{kl}(t) + s > 1 \qquad (3.7.)$$

si stabila în regiunea de saturatie negativa daca:

$$\sum_{C(k,l)\in N_r(i,j)} A_{ij,kl}\, y_{kl}(t) + s < -1 \qquad (3.8.)$$

Celule active – este considerata activa o celula a carei iesire se schimba la un moment dat. O celula activa este tot timpul în regiunea liniara sau (în cazul nostru) trece dintr-o regiune de saturatie în alta.

Analiza unui regim tranzitoriu:
La inceput, sunt câteva celule active. În caz contrar, nu exista regim tranzitoriu. În timpul regimului tranzitoriu o celula poate deveni activa daca si numai daca unul dintre vecinii directi o activeaza. Daca o celula este activa, iesirea sa se schimba în timp, adica termenul

$$\sum_{C(k,l)\in N_r(i,j)} A_{ij,kl}\, y_{kl}(t) + s \qquad (3.9.)$$

determinat de vecini se schimba. Prin urmare, o celula inactiva nu poate deveni activa daca nu are nici un vecin activ.
La sfârsitul regimului tranzitoriu, toate celulele sunt inactive.

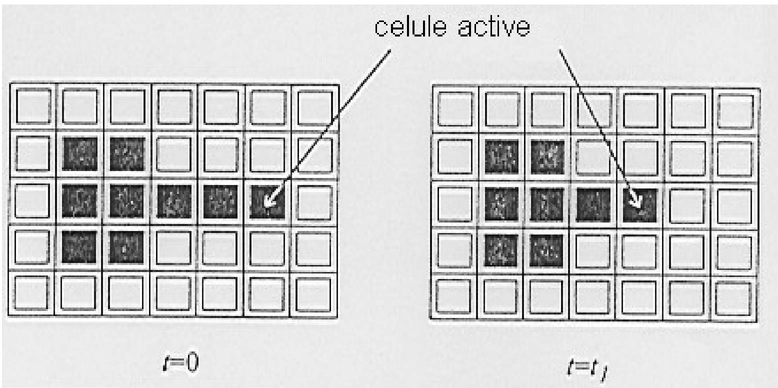


Fig. 15: Exemplu de regim tranzitoriu

Familie de masti– se refera la propagarea condusa prin mai multe masti.
Masca de activare binara – acea masca ce activeaza celula din pozitia centrala. În acest caz, unda de propagare se poate deplasa în orice directie.

Exemplul 1:

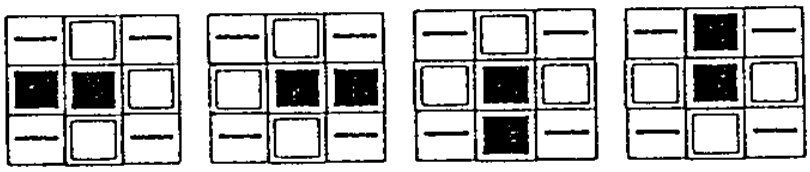


Fig. 16: Familie de masti

Regula corespunzatoare familiei de pattern-uri este aceea ca: pe o vecinatate tetraconectata sa se schimbe pixelul din mijloc daca exista cel putin trei vecini albi.

Exemplu 2: propagare condusa printr-o singura masca – directia de propagare este determinata strict de masca de activare binara (vezi figura de mai jos).

Fig. 17: Propagare condusa printr-o singura masca

Regula corespunzatoare propagarii conduse printr-o singura masca: data fiind masca de activare binara, sa se schimbe pixelii negrii care au în vecinatatea lor aceeasi masca ca si cel de fata (asimetria determina directia de propagare). Pentru masca din Fig. 17, este valabila propagarea ilustrata în Fig. 15. Masca respectiva serveste la deducerea template-urilor cu ajutorul carora se realizeaza stergerea liniilor orizontale intr-o imagine binara.

Propagare simetrica – propagare asimetrica:
O propagare este simetrica daca afecteaza pixelii negrii în acelasi mod în care îi afecteaza pe cei albi.

Exemplu pentru regula 3: Celulele albe se schimba în celule negre daca au cel putin trei vecini negri. Se aplica regula si pentru celulele negre care au cel putin trei vecini albi. Acesta este un exemplu de propagare simetrica.

Un template de propagare este simetric daca z este nul.

Propagare controlata – propagare necontrolata:
În cazul propagarii controlate intrarea contine o *imagine-masca*. Aceasta imagine-masca limiteaza propagarea. De exemplu, propagarea poate trece

peste zonele care sunt negre în imaginea de intrare. Daca propagarea este controlata, template-ul B este nenul.

Etapele de proiectare pentru template-uri de tip propagare sunt:

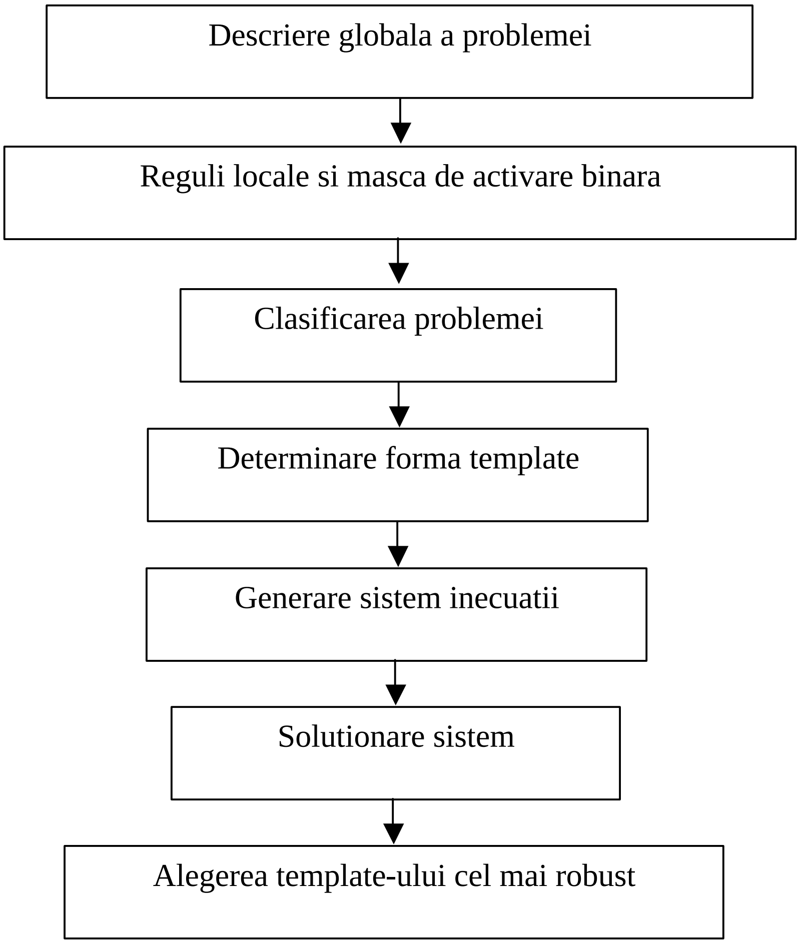


Fig. 18: Etapele de proiectare pentru template-uri de tip propagare

Descrierea globala a problemei consta în descrierea verbala a problemei în asa fel încât pentru un semnal de intrare arbitrar sa se poata deduce iesirea.
Reguli locale si masca de activare binara – deducerea regulilor locale din descrierea globala.
Clasificarea problemei stabileste daca propagarea este arbitrara sau orientata, simetrica sau nesimetrica, controlata sau necontrolata.
În cadrul determinarii formei template-ului, a_{00} se adopta întotdeauna ca parametru independent. Se noteaza cu a. Pixelii care nu conteaza "ocupa" un parametru separat. Pixelii albi si cei negri au semne opuse.
Daca propagarea este controlata (template controlat) trebuie introdusi noi parametrii independenti la template-ul B, conform cu masca de activare. În cele mai multe cazuri, este vorba de un singur parametru independent în centrul template-ului B. Uneori însa pot fi mai multi în vecinatatea pixelului central.

Daca procesarea este asimetrica (propagare asimetrica), parametrul z din template aduce un alt parametru liber.

Exemplul 1: Sa se genereze umbra de la stânga la dreapta a unui obiect negru pe o imagine binara.

Regula globala: daca exista un pixel negru într-o linie, atunci toti pixelii albi de la dreapta ar trebui sa se schimbe în pixeli negrii. Restul pixelilor ar trebui sa ramâna neschimbati.

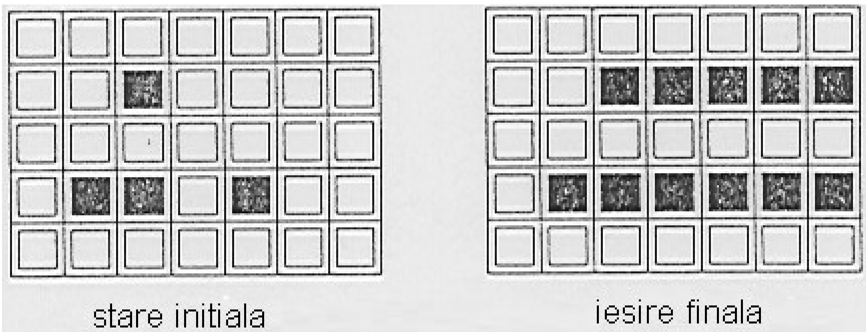


Fig. 19: Starea initiala si finala la calculul umbrei starii initiale

Reguli locale si masca de activare binara

- un pixel alb care are vecinul din stânga negru va deveni negru;
- restul pixelilor vor ramâne neschimbati.

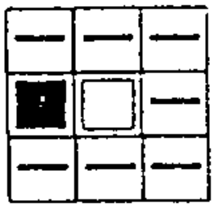


Fig. 20: Masca de activare binara

Clasificare

Propagarea este una condusa printr-o singura masca (exista doar o directie de propagare, de la stânga la dreapta). Este necontrolata (nu este nevoie de masca-imagine de selectie a obiectelor ce sufera modificari). Este asimetrica, pentru ca numai obiectele negre au umbra.

Determinarea formei template-ului

$$A = \begin{bmatrix} 0 & 0 & 0 \\ b & a & 0 \\ 0 & 0 & 0 \end{bmatrix} \quad B = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \quad z = i \qquad (3.10.)$$

Generarea sistemului de inecuatii

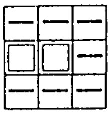
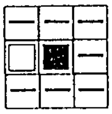
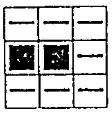
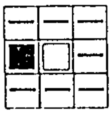
aranjarea locala a pixelilor	iesire dorita	starea	relatia
	alb	inactiv	$-a-b+i<-1$
	negru	inactiv	$a-b+i>1$
	negru	inactiv	$a+b+i>1$
	negru	activ	$-a+b+i>-1$

Fig. 21: Generarea sistemului de inecuatii

Exemplul 2: Se dau doua imagini. Prima contine câteva obiecte negre pe un fundal alb. A doua este dedusa din prima schimbând câtiva pixeli negrii în pixeli albi. În acest fel, câteva obiecte devin mai mici în cea de a doua imagine. Aceste obiecte se considera a fi marcate. Sa se proiecteze un template care sterge aceste obiecte si le lasa neschimbate pe celelalte. (Connectivity template în libraria de template-uri [51])

Regula globala: toti pixelii negrii conectati ai obiectelor marcate trebuiesc stersi.

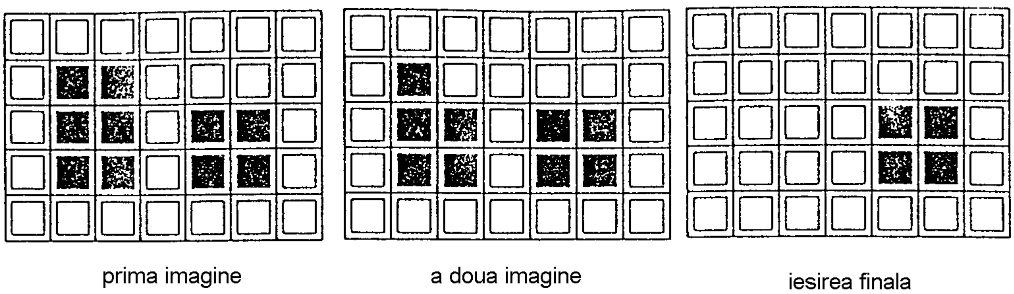


Fig. 22: Prezentarea globala a procesarii

Reguli locale si masca de activare binara

Frontul de unda se misca pe cea de a doua imagine si prima ramâne fixa. Deci punem a doua imagine pe stare, iar prima va fi pusa pe intrare.

- sa se schimbe acei pixeli negri din prima imagine care au cel putin o celula din vecinatate cu stare (iesire) alba si intrare neagra în pixeli albi;

- sa nu se schimbe restul pixelilor.

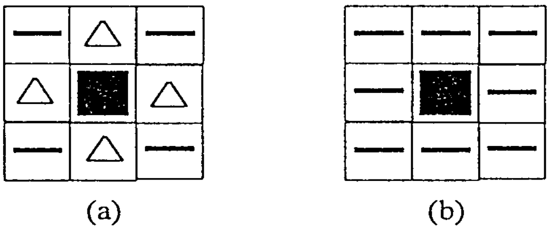


Fig. 23: Reguli locale (masti de activare)

Clasificarea
Propagarea se face în toate directiile pentru ca activarea depinde numai de numarul de pixeli care sunt identici pe intrare si pe iesire, dar nu si de pozitia lor exacta. Este controlata pentru ca propagarea afecteaza numai regiunile negre. Este asimetrica pentru ca regula se aplica numai pentru pixelii negrii.

Determinarea formei template-ului
- a_{00} este parametru liber
- delta afecteaza si pe A si pe B. Un vecin care are aceeasi intrare si iesire nu afecteaza celula. Prin urmare, forma template-ului este:

$$A=\begin{bmatrix}0 & b & 0\\ b & a & b\\ 0 & b & 0\end{bmatrix}\quad B=\begin{bmatrix}0 & -b & 0\\ -b & c & -b\\ 0 & -b & 0\end{bmatrix}\quad z=i \qquad (3.11.)$$

Celelalte etape sunt perfect analoage exemplelor prezentate pâna în acest moment.

Capitolul 4.: Aplicatii ale CNN-ului de ordinul I si template de ordin maxim r=2

În acest capitol se va prezenta un set de prelucrari de imagini realizate cu ajutorul CNN-ului, functionând atât în zona central liniara a celulelor cât si în zona de dupa saturatie a caracteristicii neliniare a celulelor. Se va începe prezentarea cu clasa de prelucrari de imagini realizata cu ajutorul CNN-ului functionând în zonele de saturatie ale caracteristicilor celulelor, la care se aplica tehnicile de proiectare de template prezentate în capitolul anterior.

Functionarea CNN-ului ca filtru spatial neliniar

În continuare se vor exemplifica o serie de prelucrari de imagini clasice sau mai putin clasice, facându-se mentiunea ca s-a dorit mai mult o exemplificare si nu o privire care sa cuprinda toate prelucrarile de imagine ce se pot realiza cu acest tip de sistem.
Exemplele de template-uri prezentate în acest capitol sunt preluate din biblioteca de template-uri "CNN Software Library" realizata de Laboratorul de programare pentru sisteme analogice si retele neuronale din cadrul Academiei ungare de stiinte, versiunea 7.3/1999, ca si cele care au servit ca exemplificare pentru tehnicile de proiectare din capitolul trecut [51-59].
Vom preleva câteva esantioane de prelucrari de imagini tipice, din cadrul urmatoarelor clase:

- prelucrari de imagini de baza;
- morfologie matematica;
- prelucrari de imagini binare;

Prelucrari de imagini de baza

Detector de componente conectate pe diagonala

Template-ul utilizat pentru realizarea acestei prelucrari de imagini este dat în relatia de mai jos:

$$A = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & -1 \end{bmatrix} \qquad B = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \qquad z = 0 \qquad (4.1.)$$

Descrierea prelucrarii de imagine:

Se da:	imaginea binara statica P
Intrare:	U arbitrar
Stare initiala:	X(0)=P
Conditii de granita:	zero pentru toate celulele virtuale
Iesirea:	Y(t)->Y(inf) este o imagine binara care arata numarul de componente conectate pe diagonala în P. Pe baza acestei informatii s-a realizat recunoasterea de caractere (OCR)

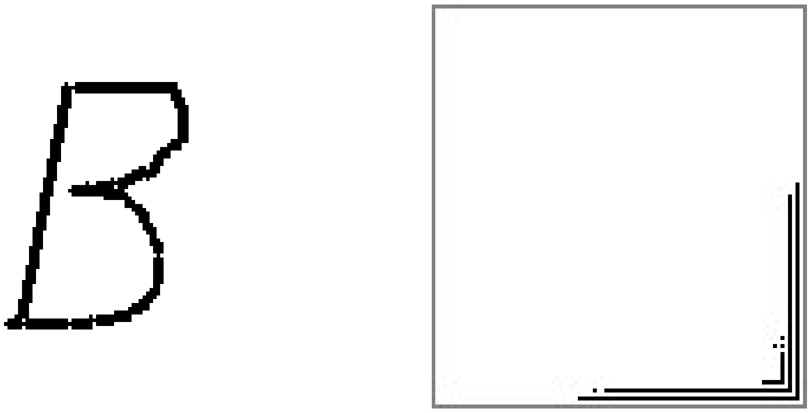


Fig. 24 a) Imagine sursa (initializare pe stare)
b) Imagine dupa prelucrare (stare finala)

Detector de contururi pe imagini binare

Template-ul utilizat pentru realizarea acestei prelucrari de imagini este dat în relatia de mai jos:

$$A = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix} \qquad B = \begin{bmatrix} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & -1 & -1 \end{bmatrix} \qquad z = -1 \quad (4.2.)$$

Descrierea prelucrarii de imagine:

Se da:	imaginea binara statica P
Intrare:	U(t)=P
Stare initiala:	X(0)=arbitrar
Conditii de granita:	zero pentru toate celulele virtuale
Iesirea:	Y(t)->Y(inf) este o imagine binara în care singurii pixeli negrii apartin marginii obiectului

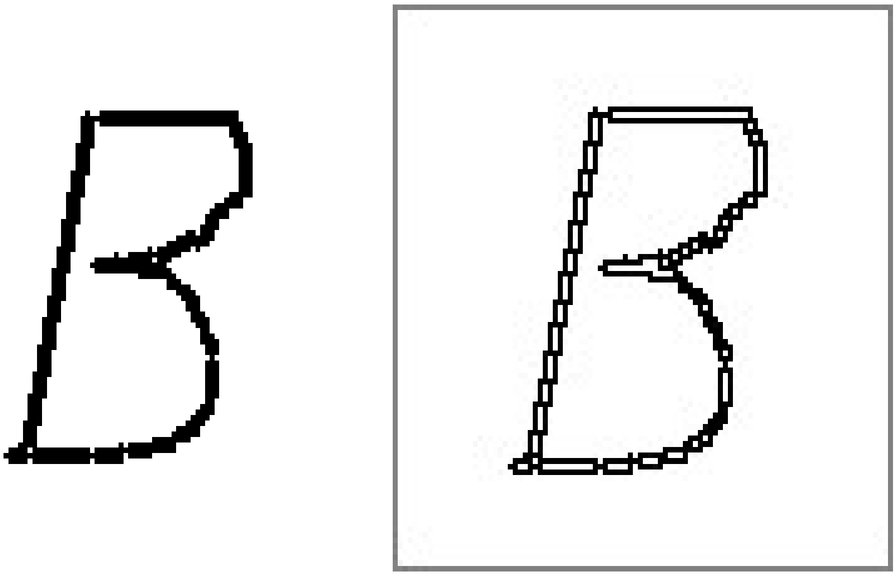


Fig. 25 a) Imagine sursa (initializare pe intrare)
b) Imagine dupa prelucrare (stare finala)

Sterge pixeli negri izolati

Template-ul utilizat pentru realizarea acestei prelucrari de imagini este dat în relatia de mai jos:

$$A = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix} \qquad B = \begin{bmatrix} 1 & 1 & 1 \\ 1 & 8 & 1 \\ 1 & 1 & 1 \end{bmatrix} \qquad z = -1 \qquad (4.3.)$$

Descrierea prelucrarii de imagine:

Se da:	imaginea binara statica P
Intrare:	U(t)=P
Stare initiala:	X(0)=arbitrar
Conditii de granita:	zero pentru toate celulele virtuale
Iesirea:	Y(t)->Y(inf) este o imagine binara în care singurii pixeli negrii apartin marginii obiectului

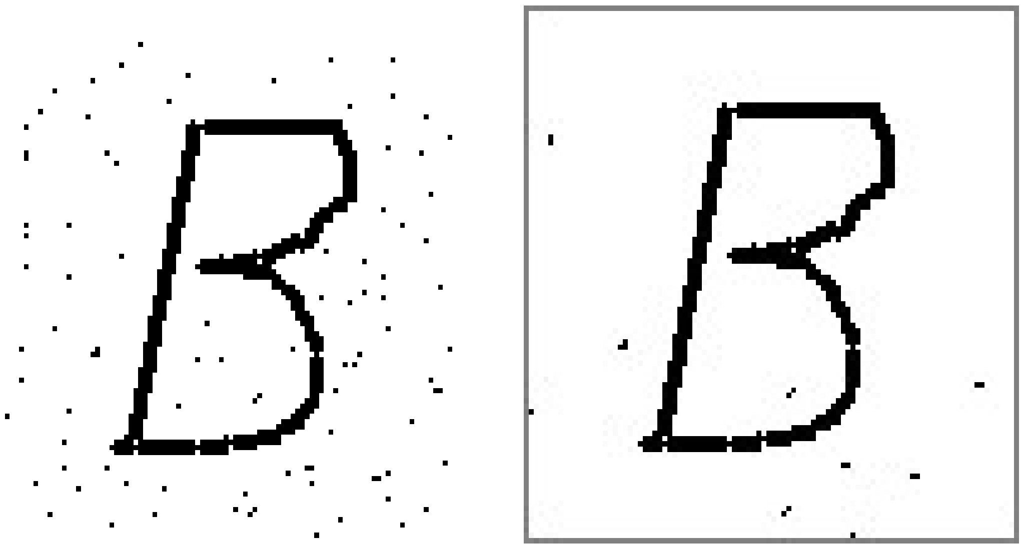


Fig. 26 a) Imagine sursa (initializare pe intrare)
b) Imagine dupa prelucrare (stare finala)

Observatii: În cadrul acestui tip de prelucrare se poate imagina un algoritm pentru a se elimina zgomotul de tip sare si piper cu ajutorul acestui tip de template si cu ajutorul unui template care calculeaza negativul unei imagini

Morfologie matematica binara

Dilatare

Template-ul utilizat pentru realizarea acestei prelucrari de imagini este dat în relatia de mai jos:

$$A = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \qquad B = \begin{bmatrix} 0 & 0 & 0 \\ 1 & 1 & 0 \\ 0 & 1 & 0 \end{bmatrix} \qquad z = 2 \qquad (4.4.)$$

Descrierea prelucrarii de imagine:

Se da:	imaginea binara statica P
Intrare:	U(t)=P
Stare initiala:	X(0)=arbitrar
Conditii de granita:	zero pentru toate celulele virtuale
Iesirea:	Y(t)->Y(inf) este o imagine binara dilatata cu un pixel fata de imaginea de pe intrare

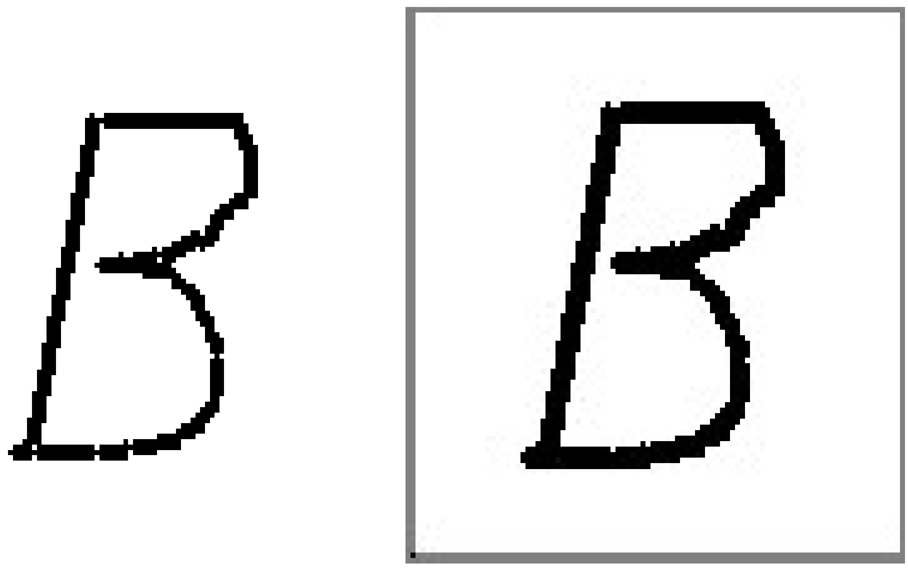


Fig. 27 a) Imagine sursa (initializare pe intrare)
b) Imagine dupa prelucrare (stare finala)

Erodare

Template-ul utilizat pentru realizarea acestei prelucrari de imagini este dat în relatia de mai jos:

$$A = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \qquad B = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 1 & 1 \\ 0 & 0 & 0 \end{bmatrix} \qquad z = -2 \quad (4.5.)$$

Descrierea prelucrarii de imagine:

Se da:	imaginea binara statica P
Intrare:	U(t)=P
Stare initiala:	X(0)=arbitrar
Conditii de granita:	zero pentru toate celulele virtuale
Iesirea:	Y(t)->Y(inf) este o imagine binara erodata cu un pixel fata de imaginea de pe intrare

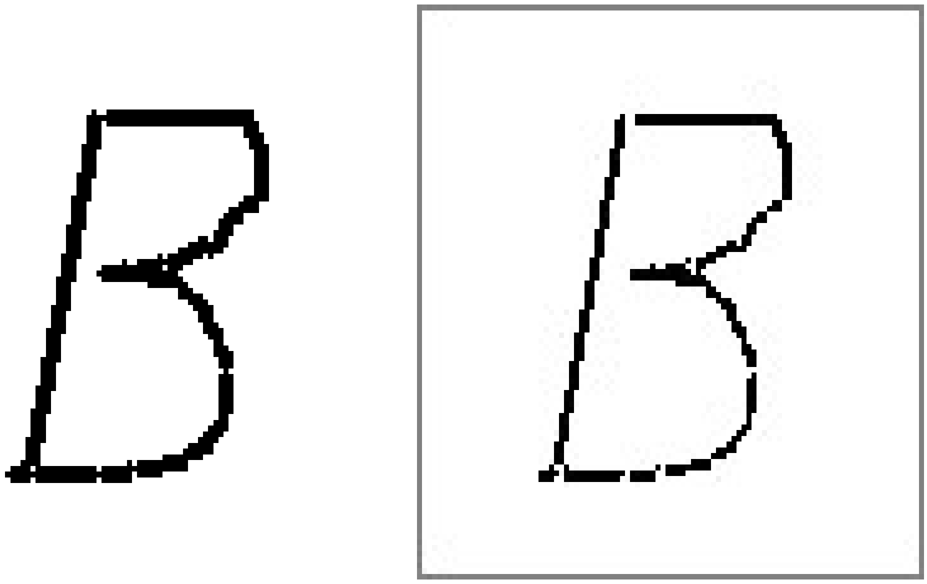


Fig. 28 a) Imagine sursa (initializare pe intrare)
b) Imagine dupa prelucrare (stare finala)

Prelucrare logica de imagini

LogicAND

Template-ul utilizat pentru realizarea acestei prelucrari de imagini este dat în relatia de mai jos:

$$A = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 0 \end{bmatrix} \qquad B = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix} \qquad z = -1 \qquad (4.6.)$$

Descrierea prelucrarii de imagine:

Se da:	imaginea binara statica P1 si P2
Intrare:	U(t)=P1
Stare initiala:	X(0)=P2
Conditii de granita:	zero pentru toate celulele virtuale
Iesirea:	Y(t)->Y(inf) este o imagine binara care corespunde rezultatului aplicarii operatorului logic SI asupra lui P1 si P2

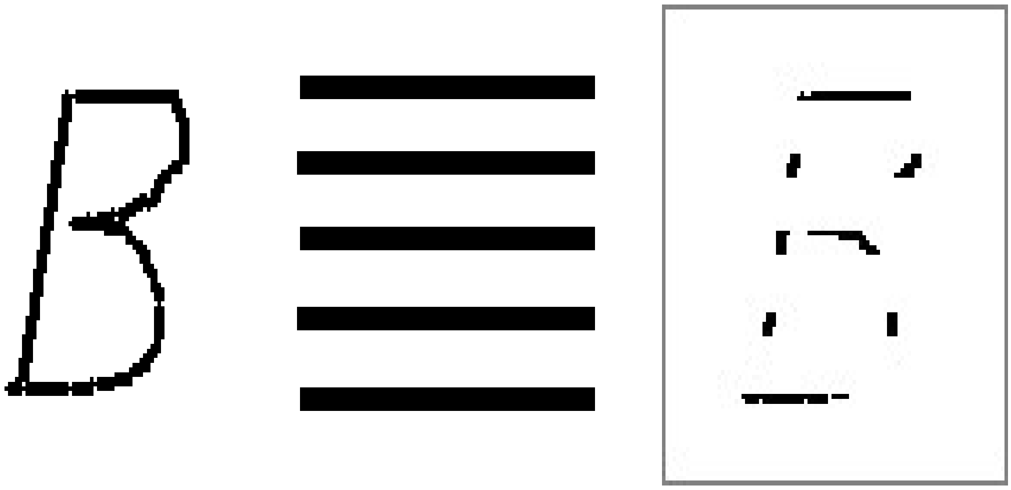


Fig. 29 a) Imagine sursa1 b) Imagine sursa2
b) Imagine dupa prelucrare (stare finala)

LogicNOT

Template-ul utilizat pentru realizarea acestei prelucrari de imagini este dat în relatia de mai jos:

$$A = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix} \qquad B = \begin{bmatrix} 0 & 0 & 0 \\ 0 & -2 & 0 \\ 0 & 0 & 0 \end{bmatrix} \qquad z = 0 \quad (4.7.)$$

Descrierea prelucrarii de imagine:

Se da:	imaginea binara statica P
Intrare:	U(t)=P
Stare initiala:	X(0)=arbitrar
Conditii de granita:	zero pentru toate celulele virtuale
Iesirea:	Y(t)->Y(inf) este o imagine binara care corespunde rezultatului aplicarii operatorului logic NEGARE asupra lui P

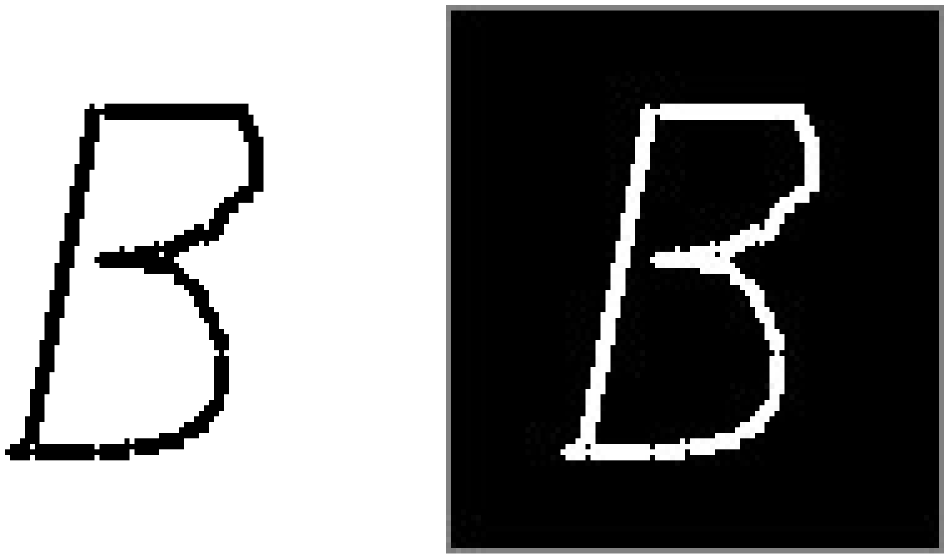


Fig. 30 a) Imagine sursa (initializare pe intrare)
b) Imagine dupa prelucrare (stare finala)

În cele ce urmeaza se prezinta un exemplu pentru functionarea CNN-ului ca filtru liniar spatial

Functionarea CNN-ului ca filtru spatial liniar

Exemplu de proiectare al unui filtru trece jos [28,29]
S-au folosit în acest exemplu template-uri simetrice, care conduc la functii de transfer reale. Template-urile de mai jos sunt obtinute printr-un proces de minimizare a unei functionale care depinde de 12 parametri (se lucreaza cu template-uri cu simetrie circulara), Procedeul este preluat de la sinteza filtrelor liniare. În acest caz locul timpului este luat de spatiu si se vorbeste despre frecvente spatiale. Evident, este vorba despre sisteme liniare stabile.

```
A=[-0.1137 -0.4549 -0.6823 -0.4549  -0.1137;
   -0.4549  0.3399  1.5896  0.3399  -0.4549;
   -0.6823  1.5896 -6.5380  1.5896  -0.6823;
   -0.4549  0.3399  1.5896  0.3399  -0.4549;
   -0.1137 -0.4549 -0.6823 -0.4549  -0.1137];
```

```
B=[0.0515 0.2059 0.3089 0.2059 0.0515;
   0.2059 0.3623 0.3127 0.3623 0.2059;
   0.3089 0.3127 0.7136 0.3127 0.3089;
   0.2059 0.3623 0.3127 0.3623 0.2059;
   0.0515 0.2059 0.3089 0.2059 0.0515];
```

Modulul functiei de transfer de tip trece-jos pentru template-ul dat mai sus:

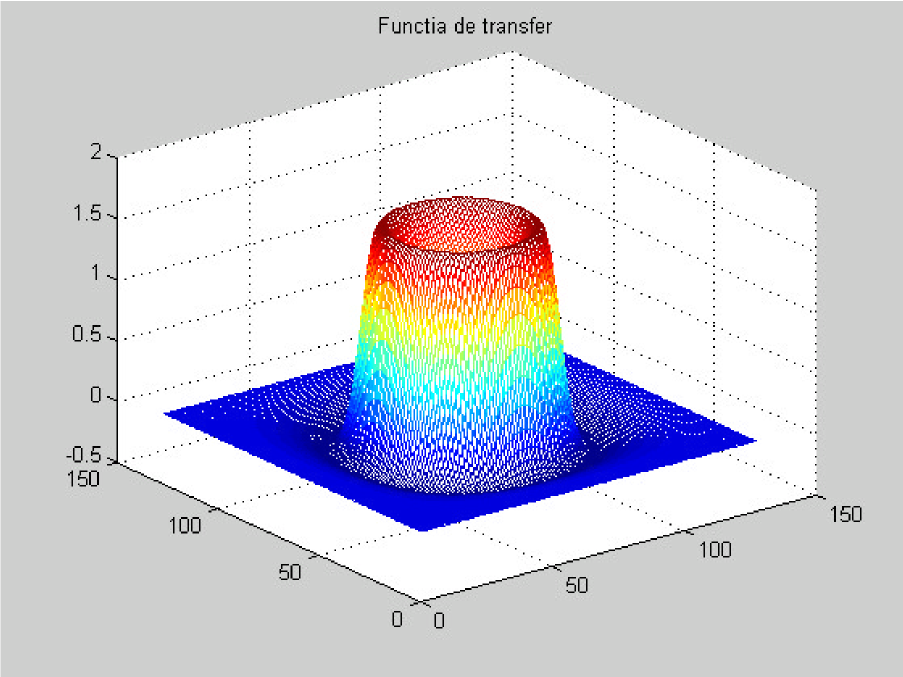


Fig. 31: Exemplu de filtru trece jos

În sectiune, caracteristica de modul se prezinta mai jos:

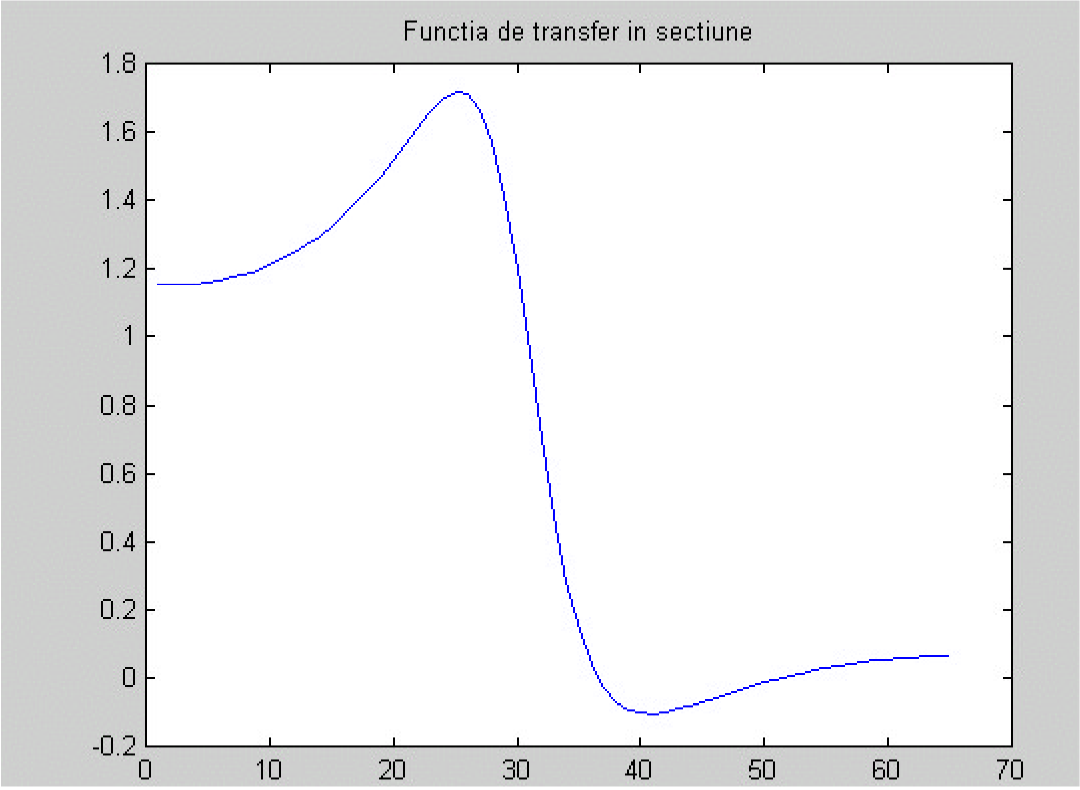


Fig. 32: Exemplu de filtru trece jos (sectiune)

Capitolul 5.: Programarea Masinii Universale de Calcul în limbajul Alpha

În cele ce urmeaza se trece în revista versiunea 3.1 a compilatorului pentru limbajul Alpha, un limbaj orientat procedura [52-59]. Este proiectat astfel încât sa deserveasca diferite medii de programare, diferite chip-uri care implementeaza conceptul de CNN si sa poata simula un sistem cu CNN.

Elemente de limbaj

Setul de caractere

Se utilizeaza setul de caractere al limbii engleze

Simboluri

Simbolurile sunt caracterele sau grupurile de caractere cu semnificatie speciala. Ele pot fi delimitatori sau operatori.

Cuvinte rezervate

Cuvintele rezervate ale limbajului pot fi scrise atât cu litere mari, câ si cu litere mici, dar odata ce s-a optat pentru una din variante, ea trebuie pastrata pe tot parcursul programului. Cuvintele rezervate nu pot fi folosite ca identificatori. Cuvintele rezervate folosite în aceasta versiune sunt listate mai jos:

ALL, ANALOG, AND, AND1NOT2, ANDNOT12, ARRAY, A_CHIP,
BEGIN, BIN, BINARY, BOOLEAN, BY, BYTE,
CONSTANT,
ELSE, END, ENDBOARD, ENDCHIP, ENDCONST, ENDFOR, ENDFUNCT, ENDIF,
FALSE, FOR, FROM, FUNCTION,
GRAY,
IF, IMAGES, INTEGER,
LAM, LLM,
NIL, NOT,
OF, OPERATIONS, OR, OR1NOT2, ORNOT12, OVERLAP,
POINTER, PROCESS, PROGRAM,
REAL, REPEAT, RETURN,
SCALARS, SEGMENTS, SKIP,
THEN, TO, TRUE,
UNTIL, USE,
WHILE, WITH,
XOR

Identificatori

Identificatorii sunt utilizati pentru:

- numele programelor;
- fisiere;
- elemente de program;
- variabile, etc.

Regulile pe care trebuie sa le îndeplineasca un identificator sunt:

- primul caracter este tot timpul o litera;
- în interiorul numelui identificatorului se pot utiliza cifre si caracterul “underscore”
- se pot utiliza litere mici si mari;
- cuvintele rezervate nu pot fi identificatori;
- exista doua cazuri speciale în care identificatorii pot contine punct si anume dupa constructiile **CHIP_SET** si **OPERATIONS FROM**. În aceste cazuri acei identificatori vizeaza nume de fisiere, iar numele fisierului este despartit prin punct de extensie (identificatori de fisiere);
- numarul maxim de caractere dintr-un identificator este de 64.

Valori exprimate literal

Constantele pot fi exprimate literal. Limbajul Alpha permite urmatoarele tipuri de valori exprimate literal:

- întregi cu sau fara semn;
- numere reale cu sau fara semn;
- valorile logice ale lui **TRUE** si ale lui **FALSE**;
- siruri de caractere literale, utilizate numai ca parametrii ai procedurilor implicite (sirurile de caractere pot contine caractere alfanumerice, punct si slash (pentru constructia numelor de fisiere si aceste valori exprimate literal trebuie încadrate de ghilimele);

Valorile exprimate literal se pot utiliza pentru definitii de constante, atribuiri, bucle. Ca si parametrii de procedura (pentru proceduri predefinite), valorile exprimate literal nu sunt acceptate.

Comentarii

Comentariile trebuie puse între /* si */. Comentariile se pot insera oriunde, cu exceptia interiorului unui cuvânt cheie. Deasemenea, nu pot depasi o linie.

Tipuri de date

Tipurile de date acceptate de limbajul Alpha sunt prezentate pe scurt dupa cum urmeaza:

- date simple (variabile scalare de tip întreg, real si binar);
- agregate ca tablourile de valori de tipurile enumerate mai sus;
- date reprezentând valori analogice, în cea mai mare parte tablouri de valori;
- tipuri agregate speciale, reprezentând instructiuni analogice, ca si elemente accesate de procedurile predefinite.

Declaratii

Declaratii de tipuri

Nu se pot face în versiunea curenta a limbajului

Declaratii de constante

Declaratiile de constante sunt utilizate pentru a da valori alfanumerice valorilor numerice sau altor valori utilizate în textul programului. Constantele sunt declarate în sectiunea de declaratii de constante dupa cum urmeaza:

```
CONSTANT
<identificator1>=<valoare1>;
<identificator2>=<valoare2>;
…
ENDCONST
```

Valorile constantelor pot fi întregi, reale cu sau fara semn sau respectiv valori logice. Utilizarea constantelor este în prezent limitata într-o oarecare masura, dar pot fi utilizate în atribuiri, expresii, structuri de control în bucla si ca argumente ale functiilor predefinite. Un exemplu al sectiunii în care se fac declaratii de constante poate fi:

```
CONSTANT
   alb=-1;
   negru=1;
   PI=3.14;
ENDCONST
```

Declaratii de variabile

Declaratiile de variabile servesc pentru definirea spatiului de stocare a valorilor scalare si agregate, utilizate de program. Variabilele se pot declara fie pe chip (real sau simulat), fie legate de placa de baza.

Variabilele declarate pe chip sunt de doua categorii:

- variabile scalare, care sunt variabile globale pentru toate unitatile de procesare;
- imagini, adica variabile locale care au forma agregata, în care elementele tabloului apartin unul câte unul unei unitati de procesare anume.

Variabilele declarate pe placa pot fi de asemenea scalari sau imagini. Variabilele alocate pe placa se pot împarti în doua grupuri:

- variabile reprezentate similar cu cele alocate pe chip (reprezentate la nivel jos ca memorii analogice locale (LAM), memorii logice locale (LLM), memorii analogice globale (GAM) si memorii logice globale (GLM);
- variabile nereprezentate pentru moment, imagini (tablouri) de dimensiuni mari si variabile de tip pointer asociate acestora.

Declaratii ale variabilelor legate de chip

Variabilele alocate pe chip sunt declarate în sectiunea alocata chip-ului în program.
Declaratia de variabile este indicata de cuvântul cheie **SCALARS** în cadrul sectiunii legate de chip. Variabilele pot fi reale, întregi sau logice. Acestea sunt întotdeauna variabile scalare si în functie de tip sunt reprezentate la nivel AMC (**A**nalog **M**acro **C**ode) ca si variabile de tip GAM sau GLM.
Declaratia de variabila de tip agregat este indicata de cuvântul cheie **IMAGES** în cadrul sectiunii legate de chip. Aceste variabile sunt fie de tipul analogic, fie de tipul logic si sunt de fiecare data tablouri. Sunt alocate la nivel hardware în memorii fie de tipul LAM, fie LLM, depinzând de tipul de date: analog sau logic. Dimensiunea tabloului este data de dimensiunea pentru care a fost realizat chip-ul. Trebuie declarata în fisierul care descrie chip-ul, care este atasat proiectului.
Descrierea sintetica a sectiunii chip-ului este data mai jos:

```
A_CHIP
SCALARS
<identifier1>: BINARY;
…
<identifier2>: INTEGER;
…
<identifier3>: REAL;
…
IMAGES
<identifier4>: BINARY;
…
<identifier5>: ANALOG;
…
ENDCHIP;
```

Un exemplu de sectiune de tip declaratie de chip este dat mai jos:

```
A_CHIP
SCALARS
   VariabilaIntreaga: INTEGER;
   VariabilaReala: REAL;
IMAGES
   ImagineBinara: BINARY;
   ImagineInTonuriDeGri: ANALOG;
ENDCHIP;
```

Declaratii ale variabilelor legate de placa de baza

Variabilele alocate pe placa de baza sunt declarate în sectiunea corespunzatoare placii de baza a programului. Declaratia de variabile este indicata de cuvântul-cheie **SCALARS** din interiorul sectiunii placii. Aceste variabile pot fi: reale, întregi sau binare. Ele sunt tot timpul variabile scalare si în functie de tipul de date cu care se lucreaza (analogic sau logic) sunt alocate ca variabile de tip GAM sau GLM.

Declaratia de tip de date agregate este indicata de cuvântul-cheie **IMAGES** în cadrul sectiunii placii. Variabilele locale sunt fie analogice, fie logice si sunt tablouri care în functie de tipul de date sunt alocate ca variabile de tip LAM sau LLM. Variabilele analogice sunt alocate în spatiul de adresare legat de ARAM (Analogic RAM), pe când toate celelalte sunt alocate în zona DRAM a chip-setului.

Declaratiile de imagini la care dimensiunea tabloului este omisa utilizeaza ca optiune implicita dimensiunea chip-ului, pe când declararea explicita a dimensiunii tabloului se recomanda pentru imagini de dimensiuni mai mari.

Imaginile de dimensiuni mai mari pot fi doar de tip binar sau în tonuri de gri. Imaginile analogice pot fi doar de dimensiunea pentru care a fost construit chip-ul.

Dimensiunile tablourilor pot fi definite fie utilizând valori literale, fie constante.

Descrierea sintetica a sectiunii placii este data mai jos:

```
E_BOARD
SCALARS
<identifier1>: INTEGER;
…
IMAGES
<identifier2>: BINARY;
…
<identifier3>: BYTE;
…
<identifier4>: ANALOG;
…
<identifier5>: ARRAY [literal1, constant2] OF BINARY;
…
<identifier6>: ARRAY [constant1, literal2 ] OF BYTE;
…
ENDBOARD;
```

Un exemplu de sectiune pentru placa este dat mai jos:

```
E_BOARD
   SCALARS
      VariabilaIntreaga: INTEGER;
```

```
    IMAGES
        ImagineBinaraDeDimensiuneaCipului: BINARY;
        ImagineInTonuriDeGriDeDimensiuneaCipului: BYTE;
        ImagineInTonuriDeGriDeDimensiuneaARAM: ANALOG;
        ImagineBinara: ARRAY [512, 512] OF BINARY;
        ImagineInTonuriDeGri: ARRAY [512, 512] OF BYTE;
ENDBOARD;
```

Declaratii de functii

Declaratia de functie este prezentata schematic mai jos:

```
FUNCTION <nume functie>;
USE (<nume template-uri>);
<declaratii>
…
ENDFUNCT;
```

Functiile sunt utilizate pentru structurarea programului în mod convenabil si pentru permite utilizatorului sa utilizeze template-urile existente. Versiunea aceasta a compilatorului nu permite utilizarea parametrilor, nu accepta returnarea de valori si variabile locale functiilor.
Numele unei functii este un identificator care respecta regulile acestora.
În declaratia începând cu cuvântul cheie **USE** se mentioneaza, separate de virgule, numele template-urilor ce urmeaza sa fie activate. Compilatorul genereaza, pe baza acestor nume de template-uri, instructiunile necesare pentru încarcarea acestor template-uri. În timpul generarii de cod, compilatorul gestioneaza reîncarcarea instructiunilor dupa reintoarcerea dintr-un apel de functie, pentru a restaura configuratia precedenta a template-ului.

Un exemplu de functie:

```
…
CONSTANT
    TimpRulare=5;
    LimitaInferioara=0.25;
    LimitaSuperioara=0.5;
ENDCONST
…
```

```
FUNCTION Scheletizare;
USE (Skel1, Skel2, Skel3, Skel4, Skel5, Skel6, Skel7, Skel8);

REPEAT VariabilaIntreaga := 1 TO 10 BY 1;
Skel1(LLM3, LLM3, LLM3, TimpRulare, LimitaInferioara, LimitaSuperioara,
LAM4, LAM5);
Skel2(LLM3, LLM3, LLM3, TimpRulare, LimitaInferioara, LimitaSuperioara,
LAM4, LAM5);
Skel3(LLM3, LLM3, LLM3, TimpRulare, LimitaInferioara, LimitaSuperioara,
LAM4, LAM5);
Skel4(LLM3, LLM3, LLM3, TimpRulare, LimitaInferioara, LimitaSuperioara,
LAM4, LAM5);
Skel5(LLM3, LLM3, LLM3, TimpRulare, LimitaInferioara, LimitaSuperioara,
LAM4, LAM5);
Skel6(LLM3, LLM3, LLM3, TimpRulare, LimitaInferioara, LimitaSuperioara,
LAM4, LAM5);
Skel7(LLM3, LLM3, LLM3, TimpRulare, LimitaInferioara, LimitaSuperioara,
LAM4, LAM5);
Skel8(LLM3, LLM3, LLM3, TimpRulare, LimitaInferioara, LimitaSuperioara,
LAM4, LAM5);
ENDREPEAT;

ENDFUNCT;
```

Scop si vizibilitate

Din punctul de vedere al scopului si vizibilitatii toate variabilele sunt pentru aceasta versiune globale. Aceasta este o facilitate care pe masura ce limbajul va evolua, se va dezvolta.

Specificatii legate de interfata

Limbajul de programare Alpha utilizeaza urmatoarele tipuri de interfete externe:

- interfata legata de argumentele care pot fi pasate programului;
- definitia chip-ului;
- interfata cu definitia template-ului.

Argumentele pasate programului

Argumentele programului servesc la definirea fisierelor externe utilizate pentru încarcarea de imagini de intrare si de iesire. De asemenea, fisierele cu parametrii care descriu o anume implementare pot fi argumente.
Aceste entitati sunt fisiere logice, fisierele fizice urmând sa fie atasate la run-time înainte de executia propriu-zisa a programului compilat.
Este posibila utilizarea numai a unui set de parametrii, dar trebuie lasat un spatiu pentru cei lipsa, daca dupa ei mai urmeaza parametrii, ca în exemplele ce vor urma.
Formal, pasarea argumentelor se face în felul urmator:

```
PROGRAM <identificator>
([<imagini de intrare>]; [<imagini de iesire>]; [<fisiere cu parametrii>]);

Exemple de cod:

PROGRAM exemplu(in; out; param);
PROGRAM exemplu(in; out);
PROGRAM exemplu(in);
PROGRAM exemplu( ; out; param);
PROGRAM exemplu( ; out);
```

Declararea chip-ului si a placii

Compilatorul lucreaza cu programe scrise în Alpha pentru diferite implementari hard si medii integrate. Mediul tinta (hard) trebuie descris într-un fisier de definitie. Forma declaratiei este urmatoarea:

```
CHIP_SET <nume fisier definitie>;
```

Numele fisier definitie poate fi orice nume (fara cale), continând un singur identificator sau un identificator continând nume ti extensie separate de punct. Numele nu se verifica de catre compilator.
Fisierul este unul de descriere a chip-ului si este format din perechi nume-valori. Scopul sau este definirea facilitatilor speciale ale chip-ului.

Un exemplu de declaratie de chip este:

```
CHIP_SET cp4000.eng
```

Un exemplu de fisier care descrie chip-ul si placa este dat mai jos:

```
/* SimCNN CNN-UM simulator */

/* Size of the chip template register file */
TEM_LIMIT: 64;

/* Limits for the number of image memories */
LAM_LIMIT: 32;
LLM_LIMIT: 32;

/* Lower and upper limits for the DRAM allocated LAMs, LLMs */
CHS_LAM_LOW: 33;
CHS_LAM_LIMIT: 63;
CHS_LLM_LOW: 33;
CHS_LLM_LIMIT: 63;
```

```
/* Limits for the chip global variables */
GAM_LIMIT: 512;
GLM_LIMIT: 512;

/* Lower and upper limits for the DRAM allocated GAMs, GLMs */
CHS_GAM_LOW: 513;
CHS_GAM_LIMIT: 1024;
CHS_GLM_LOW: 513;
CHS_GLM_LIMIT: 1024;

/* Local iput */
TEM_INPUT: LAM(1-32), LLM(1-32);

/* Local initial state */
TEM_INIT: LAM(1-32), LLM(1-32);

/* Local Fixed state */
TEM_FIXED: LLM(1-32);

/* Local bias map */
TEM_BIAS:  LAM(1-32), LLM(1-32);

/* Local output image */
TEM_OUT: LAM(1-32), LLM(1-32);

/* Valid logical operations */
LOG_OPS:  AND, OR, NOT, NAND, NOR, XOR, EQU, OR1NOT2,
ORNOT12, AND1NOT2, ANDNOT12;

/* Operands and target of logical operations */
LOG_OP1: LLM(1-32);
LOG_OP2: LLM(1-32);
LOG_OUT: LLM(1-32);

/* Valid arithmetic operations */
ARIT_OPS: +,-,*;

/* Operands and targets of arithmetic operations */
ARIT_OP1: LAM(1-32);
ARIT_OP2: LAM(1-32);
ARIT_OUT: LAM(1-32);

/* Operands and targets of conversion type operations */
CNV_SOURCE: LAM(1-32), LLM(1-32);
CNV_TARGET: LAM(1-32), LLM(1-32);
CAL_SOURCE: LAM(1-32), LLM(1-32); /* Calculate LAM,LLM numeric,
logic features */
FILL_TARGET: LAM(1-32), LLM(1-32); /* Fill from GAM, GLM contents */
```

```
/* Operands and targets of move type operations */
MOV_SOURCE: LAM(1-63), LLM(1-63);
MOV_TARGET: LAM(1-63), LLM(1-63);

/* Targets and sources of image load operations */
HOST_LOAD: LAM(33-63);
HOST_SAVE: LAM(33-63);

/* Sources of image display operations */
HOST_DISP: LAM(33-63);
```

Declaratia de template

Declaratia de template are forma:

```
OPERATIONS FROM <nume fisier template>;
```

Constructia de mai sus serveste la definirea fisierului în care numele template-urilor sunt listate. Se pot utiliza în program doar template-urile listate în acest fisier. Numele fisierului respecta aceleasi reguli ca la declararea chip-ului si a placii.
Un exemplu de declaratie:

```
OPERATIONS FROM example.tms
```

Un exemplu de fisier este dat mai jos:

```
glife1
glife2
```

Fisierul arata ca se vor utiliza cele doua template-uri care simuleaza jocul vietii glife1 si respectiv glife2.

Expresii

Expresiile pot fi de doua feluri:

- expresii logice si aritmetice între imagini;
- expresii logice si aritmetice între scalari.

Pentru ambele tipuri de expresii, numarul operanzilor este limitat la doi.

Expresii între imagini

Utilizatorul poate scrie expresii între imagini utilizând operatiile aritmetice si logice: adunarea (+), înmultirea (*), scaderea (-), AND, OR, NOT, NAND, NOR, XOR, OR1NOT2 (OR cu un operand doi negat), ORNOT12 (OR cu un operand

unu negat), AND1NOT2 (AND cu un operand doi negat), ANDNOT12 (AND cu un operand unu negat).
Partea din dreapta a expresiei poate sa aiba fie numai imagini de tip chip, fie numai de tip placa. De exemplu:

```
ImagineBinara3 := ImagineBinara1 OR ImagineBinara2;
```

Expresii numerice

Expresiile numerice pot fi ori întregi, ori reale. Se utilizeaza adunarea, înmultirea, scaderea si expresia este limitata la doi operanzi.

Expresii logice

Pentru moment expresii logice ale variabilelor scalare nu sunt suportate în atribuiri.

Structura unui program scris în limbaj Alpha

Programul principal

Forma generala este data mai jos:

```
PROCESS <nume>;
USE (<nume template-uri>);
<declaratii>
…
ENDPROCESS;
```

Programul principal se aseamana întrucâtva cu declaratia de functie. În acelasi timp exista diferente. Numele nu are prea mare importanta, fiind dat doar din ratiuni informative. Declaratia care începe cu USE defineste template-urile care vor fi utilizate în programul principal.
Un exemplu are forma de mai jos:

```
PROCESS Exemplu;
USE (Dilation, Erosion);
   HostLoadPic(in, BufferImagineBinara);
   ImagineBinara1 := BufferImagineBinara;
   Dilation(ImagineBinara1, ImagineBinara1, ImagineBinara2, TimpRulare,
LimitaInferioara,
       LimitaSuperioara, ImagineBinara4, ImagineBinara6);
   Erosion(ImagineBinara2, ImagineBinara2, ImagineBinara1, TimpRulare,
LimitaInferioara,
       LimitaSuperioara, ImagineBinara4, ImagineBinara6);
   BufferImagineBinara := ImagineBinara1;
```

```
    HostSavePic(out, BufferImagineBinara);
ENDPROCESS;
```

Instructiuni care se executa

Instructiunea FOR-WITH

Instructiunea FOR-WITH are un rol special, ea asigurând procesarea unor imagini de dimensiuni mai mari decât marimea chip-ului. Ea determina compilatorul sa genereze codul corespunzator situatiei când imaginea este împartita în sub-imagini de marimea celei care "încape" pe chip.

```
FOR ( <lista de identificatori #1> ) BY <mod de procesare>
WITH ( <lista de identificatori #2> )
OVERLAP ( <valoare intreaga>, <valoare intreaga>)
<apeluri de functii, proceduri>
ENDFOR;

FOR ( <lista intrare #1>; <lista iesire #1> ) BY <mod de procesare>
WITH ( <lista de intrare #2>; <lista de iesire #2> )
OVERLAP ( <valoare intreaga>, <valoare intreaga>)
<apeluri de functii, proceduri>
ENDFOR;
```

Numele imaginilor mai mari decât dimensiunea chip-ului care urmeaza sa fie procesate (fie de intrare, fie de iesire) sunt listate în <lista de identificatori #1>. Fiecare dintre imaginile mai mari decât dimensiunea chip-ului trebuie sa aiba o imagine alocata pe placa, de dimensiunea chip-ului. Memoriile de lucru sunt listate în <lista de identificatori #2>. Sunt doua posibilitati de codare pentru listele de identificatori:

- daca lista de identificatori este o secventa unica de nume separate prin virgule, fiecare dintre imaginile în corespondenta cu numele pot fi atât operanzi de intrare cât si operanzi de iesire. În acest caz un segment din fiecare imagine de dimensiune mare este "taiat" si „lipit" din nou în fiecare din ciclurile de procesare;
- daca lista de identificatori este partitionata de punct si virgula în subliste de intrare si de iesire, imaginile de intrare sunt utilizate numai pentru taierea de segmente si numai imaginile de iesire sunt utilizate pentru operatii de lipire a segmentelor de imagine în fiecare din ciclurile de procesare.

Lungimile sublistelor de intrare si de iesire si numarul membrilor sublistei corespunzatoare imaginilor de dimensiunea chip-ului trebuie sa fie egale.

Modul de procesare se va defini mai târziu. Pentru moment, poate fi doar segmentat, în care imaginile mai mari se împart în segmente de marimea chip-ului. Segmentele pot fi definite tinând cont de suprapunerea lor sau nu. Suprapunerea are valoare întreaga, deoarece este în esenta numarul comun de pixeli ce apartin de doua segmente.

Sectiunea <apeluri de functii, proceduri> reprezinta o secventa de instructiuni de procesare, în care fiecare imagine de dimensiune mare poate fi folosita ca si operand sau imagine de stocare a rezultatului si mai mult, ca parametru al procedurilor predefinite.
Nu se recomanda ca, în interiorul unei instructiuni FOR sa se utilizeze memorii cu capacitate de stocare de dimensiunea chip-ului chiar daca în acest moment acest lucru este permis.
O constrângere foarte puternica este aceea ca instructiunile FOR se pot utiliza doar în programul principal si nu se pot utiliza recursiv. Mai mult, este recomandat sa se utilizeze cu FOR doar acele instructiuni sau proceduri predefinite care nu lucreaza cu imagini mari.
Instructiunea FOR nu este permisa în toate implementarile.

Atribuirea

Are forma:

```
<variabila> := <expresie>;
```

Operanzii într-o instructiune de atribuire pot fi fie imagini, fie scalari, fie expresii în care acestea sunt utilizate.

De exemplu:

```
BufferImagineBinara := ImagineBinara1;
```

în care entitatea din stânga poate fi:

- variabila de tipul aritmetic sau logic local (pe chip), dupa cum este declarat în sectiunea A_CHIP;
- variabila stocata în memorie ca imagine (binara, în tonuri de gri sau analogica) dupa cum sunt declarate în sectiunea E_BOARD;
- variabile întregi, reale, care au fost declarate fie în A_CHIP, fie în E_BOARD.

Entitatea din dreapta *trebuie sa fie de acelasi tip de date* ca cea din stânga:

- variabila de tipul aritmetic sau logic local (pe chip), dupa cum este declarat în sectiunea A_CHIP;
- variabila stocata în memorie ca imagine (binara, în tonuri de gri sau analogica) dupa cum este declarata în sectiunea E_BOARD;
- variabila întreaga, reala, care a fost declarata fie în A_CHIP, fie în E_BOARD;
- expresii aritmetice si logice construite din doi operanzi declarati fie în A_CHIP, fie în E_BOARD;
- constanta numerica de valoare întreaga sau reala;
- expresii limitate construite din variabile întregi, variabile reale, valori numerice si constante;
- anumite functii predefinite.

Instructiunea conditionala

Aceasta are forma:

```
IF <conditie>
THEN <instructiuni>
ELSE  <instructiuni>
ENDIF;

IF <conditie>
THEN <instructiuni>
ENDIF;
```

<conditie> este o expresie relationala, care este construita cu operatorii relationali (<, >, <=, >=, =, <>) si cu operanzi variabile întregi, valori literale zecimale, si din functii predefinite. Se utilizeaza în aceasta versiune numai expresii cu doi membrii.

De exemplu:

```
IF VariabilaIntreaga <> 0
THEN
Erosion(ImagineBinara4, ImagineBinara4, ImagineBinara4,
    TimpRulare, LimitaInferioara, LimitaSuperioara,
    ImagineBinara5, ImagineBinara7);
ENDIF;
```

Apelul de functie

Numele functiilor declarate pot fi utilizate ca instructiuni de tip apel functie, însa în acest moment, numai apelurile de functii fara parametrii si care nu returneaza valori sunt implementabile.

De exemplu:

```
Scheletizare;
```

Bucle

Limbajul si compilatorul suporta trei tipuri de bucle:

- bucla FOR, în care bucla lucreaza pe baza de valoare initiala, valoare finala si pas;
- bucla WHILE, în care atâta timp cât conditia este adevarata se executa corpul buclei; verificând conditia la începutul buclei;
- bucla UNTIL, în care atâta timp cât conditia este falsa se executa corpul buclei;, verificând conditia la sfârsitul buclei;

Formatul este urmatorul:

```
REPEAT FOR <variable1> := <valoare initiala> TO <valoare de sfarsit> BY <valoare pas>;
<.instructiuni>
…
ENDREPEAT;
```

Variabila este una de tip întreg, globala, celelalte fiind valori literale zecimale, constante sau variabile întregi globale.

```
REPEAT WHILE <conditie>;
<.instructiuni>
…
ENDREPEAT;

REPEAT UNTIL <conditie>;
<.instructiuni>
…
ENDREPEAT;
```

Instructiunea RETURN

Este posibila întoarcerea dintr-o functie apelata prin RETURN;.
Desigur, în momentul când se iese dintr-o functie, se genereaza automat un RETURN.

Proceduri si functii predefinite

Procedurile si functiile predefinite pot fi clasificate în urmatoarele categorii:

1. Template-uri ca proceduri predefinite speciale
2. Proceduri pentru conversie de date
3. Proceduri pentru controlul intrarii optice
4. Proceduri de manipulare de template-uri
5. Manipulare de sub-imagini
6. Servicii de timing
7. Apelul de proceduri DSP externe
8. Controlul intrarii si iesirii prin fisiere
9. Servicii aditionale în ceea ce priveste placa

Conventii în ceea ce priveste procedurile predefinite:

- tipurile permise sunt: nume logice de fisiere, valori literale de tip sir de caractere, valori întregi si reale, simboluri predefinite, imagini analogice, imagini în tonuri de gri, imagini binare;
- valorile numerice pot fi atribuite fie ca si constante declarate, fie ca variabile;
- compilatorul verifica atât numarul cât si tipul parametrilor;
- când se scrie un program, autorul trebuie sa fie familiarizat cu constrângerile impuse de chip.

Template-uri ca proceduri predefinite speciale

Aceste proceduri au sapte parametrii dupa cum urmeaza:

- imaginea de intrare – o variabila de tip imagine;
- imaginea pusa pe starea initiala - o variabila de tip imagine;
- imaginea de iesire - o variabila de tip imagine;
- timp de rulare – o valoare reala, fie constanta fie variabila;
- valoare de granita pentru intrare – o valoare reala sau ZEROFLUX sau PERIODIC;
- valoare de granita pentru iesire – o valoare reala sau ZEROFLUX sau PERIODIC;
- imaginea care mascheaza starea - o variabila de tip imagine;
- imaginea cu sursele de curent continuu - o variabila de tip imagine;

Ultimii trei parametrii se pot omite. Daca ei lipsesc, se genereaza un cod care contine aceeasi valoare pentru valoarea de granita pentru iesire ca si cea pentru intrare.
Variabilele parametrii pot fi reprezentabile sau nu (fizic) pe chip, însa cu cele nereprezentabile se lucreaza numai cu instructiunea FOR-WITH.

În cele ce urmeaza se prezinta succint tipurile de proceduri însotite de câteva comentarii.

Proceduri pentru conversie de date si pentru calculul atributelor globale ale imaginii

GlobOrPixels	calculeaza functia logica OR pentru pixelii unei imagini binare
GlobAndPixels	calculeaza functia logica AND pentru pixelii unei imagini binare
GlobCountPixels	calculeaza numarul pixelilor negrii dintr-o imagine binara Parametrii procedurii sunt:
GlobAvgPixels	calculeaza valoarea medie a pixelilor dintr-o imagine în tonuri de gri
GlobMinPixel	calculeaza valoarea minima a pixelilor dintr-o imagine în tonuri de gri
GlobMaxPixel	calculeaza valoarea maxima a pixelilor dintr-o imagine în tonuri de gri
GlobMeanPixel	calculeaza valoarea medie a pixelilor dintr-o imagine în tonuri de gri
ConvReal2Logic	converteste catre valori logice dependente de semn imaginea sursa în felul urmator: daca variabila sursa este pozitiva rezultatul va fi TRUE, altfel FALSE
ConvLogic2Real	converteste catre valori reale valorile logice în felul urmator: daca variabila sursa este TRUE atunci rezultatul va fi 1.0, altfel –1.0.

Parametrii procedurilor sunt imagini în tonuri de gri, imagini binare, variabile reale sau logice, dupa caz.

Proceduri pentru controlul intrarii optice

OptInput	initializeaza intrarea optica, procedura având un singur parametru: imaginea destinatie (o imagine alocata pe chip)

Proceduri de manipulare de template-uri

TemLoad	încarca un template de pe disk într-un registru special pentru template-uri
TemSave	salveaza un template pe disk dintr-un registru de template-uri

Manipulare de sub-imagini, proceduri pentru deplasare de date

MoveImageRow	muta o linie între doua imagini. Fiecare dintre imagini poate fi una alocata pe chip sau una reprezentabila
ExtrSubImage	extrage o imagine de dimensiunea chip-ului dintr-o imagine mare
InsSubImage	insereaza o imagine de dimensiunea unui chip într-o imagine mare
GetPixel	obtine valoarea unui pixel
PutPixel	seteaza o valoare a unui pixel

Servicii de timing

SwSetTimeStep	seteaza pasul de timp pentru iterarea evolutiei CNN

Apelul de proceduri DSP externe

DspCall	activeaza un algoritm realizat DSP

Controlul intrarii si iesirii prin fisiere

HostLoadPic	încarca o imagine dintr-un fisier de pe disk
HostSavePic	salveaza o imagine într-un fisier de pe disk
HostLoadSeq	încarca o secventa de imagini din fisiere de pe disk
HostSaveSeq	salveaza o secventa de imagini în fisiere de pe disk
HostDisplay	afiseaza o imagine în fereastra de imagine definita

Servicii aditionale în ceea ce priveste placa

HostStartGrab	starteaza modul trece-prin pentru frame-grabber
HostStopGrab	stopeaza modul trece-prin pentru frame-grabber
HostGrabImage	captureaza si stocheaza imaginea pe placa
HostMoveImage	muta o imagine sau un plan de imagine între o variabila de tip imagine si sistemul de buffere intern definit de nivelul AMC (denumite INPUT si OUTPUT). Transferul se poate face dinspre buffer si spre buffer.
HostLoadParameter	se încarca un fisier de parametrii
HostGetImage	achizitioneaza si stocheaza o imagine dintr-o fereastra de pe ecran
HostPrint	afiseaza un mesaj la dispozitivul standard de iesire

Amanunte despre parametrii tuturor procedurilor prezentate mai sus se gasesc în [52,53], dorindu-se prin prezentarea sintaxei si a procedurilor predefinite schitarea posibilitatilor de procesare de imagine existente până în acest moment în standardul ce descrie limbajul Alpha. De asemenea, în acest fel cititorul are si imaginea limitarilor impuse atât de hardware, cât si prin standardul limbajului.

Capitolul 6.: O arhitectura generala de circuit pentru celula de ordinul II

În acest capitol se prezinta o serie de rezultate referitoare la o arhitectura speciala de CNN, constând din celule de tip diport de forma celei prezentate în Fig. 33.

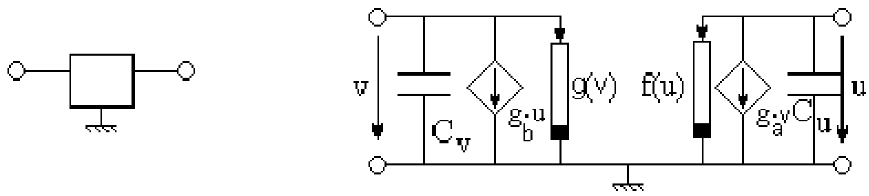


Fig 33: Structura generala a celulei de ordinul II

Sistemul de ecuatii care descrie functionarea celulei este:

$$\begin{cases} \dfrac{du(t)}{dt} = \dfrac{1}{C_u}(-f(u) - g_a v) = F(u,v) \\ \dfrac{dv(t)}{dt} = \dfrac{1}{C_v}(-g_b u - g(v)) = G(u,v) \end{cases} \tag{6.1.}$$

Termenii neliniari f(u) si g(v) pot fi în general de orice forma, dar este preferabil sa fie de tip liniar pe portiuni [60-63], ca în Fig. 34 si 35.
Prin liniarizare în jurul unui punct de functionare se obtin ecuatiile:

$$\begin{cases} \dfrac{du(t)}{dt} = g\,(f_u u + f_v v) \\ \dfrac{dv(t)}{dt} = g\,(g_u u + g_v v) \end{cases} \tag{6.2.}$$

unde f_u, f_v, g_u, g_v si parametrul γ sunt dati de relatiile:

$$f_u = -\left.\frac{\partial F}{\partial u}\right|_{(U_0,V_0)} \quad f_v = -g_a \quad g_u = -\frac{C_u}{C_v} g_b \quad g_v = -\frac{C_u}{C_v}\left(\left.\frac{\partial G}{\partial v}\right|_{(U_0,V_0)}\right) \quad g = \frac{1}{C_u} \tag{6.3.}$$

Exemple de structuri de celule particulare

Celulele de tip diport din literatura [60-87] pot fi considerate cazuri particulare ale celulei din Fig. 33.

Astfel, celula Chua [60-62] simplificata de (ordin II) are configuratia din Fig. 34:

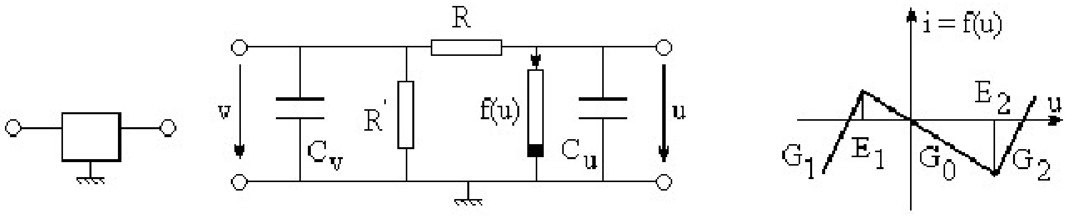


Fig 34: Celula Chua simplificata

în care f_u, f_v, g_u, g_v si parametrul γ sunt dati de relatiile:

$$f_u = -\frac{|C_u|}{C_u} G_i, \quad f_v = \frac{|C_u|}{C_u}, \quad g_u = \frac{|C_u|}{C_v}, \quad g_v = -\frac{|C_u|}{C_v}\left(1 + \frac{G'}{G}\right) \qquad g = \frac{1}{|C_u|} \quad (6.4.)$$

O alta celula propusa în [63] este prezentata în Fig. 35:

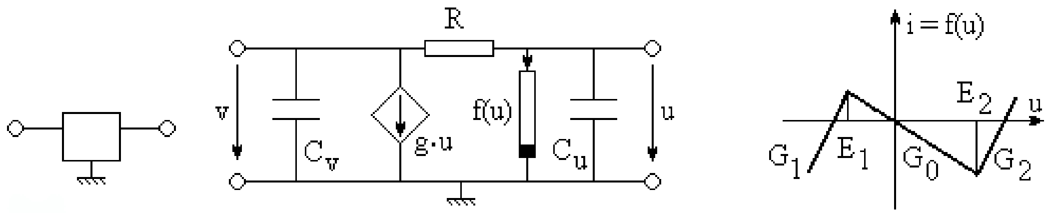


Fig 35: Versiune a celulei generale

Pentru caracteristica rezistorului neliniar prezentat în figura se da relatia dintre elementele matricii Jacobian si elementele de circuit precum si dependenta lui γ de elementele de circuit:

$$f_u = -(G + G_0) \qquad f_v = G \qquad g_u = \frac{C_u}{C_v}(G - g) \qquad g_v = -\frac{C_u}{C_v} G \qquad g = \frac{1}{C_u} \quad (6.5.)$$

Capitolul 7.: Studiul retelei neuronale celulare 1D formate din celule de ordinul II conectate cu o vecinatate de raza unitate

Prin interconectarea folosind surse comandate sau retele rezistive, pe *doua straturi,* [60-62, 64-68, 78-80] a portilor similare prezentate în capitolul precedent se obtine un sistem autonom a carui functionare în zona liniara centrala a celulelor liniare pe portiuni sa poata realiza o procesare de semnal utila.

Interconexiunile

Un mod de a realiza o procesare de semnal este de a interconecta celulele într-o retea dublu cuplata care le leaga prin intermediul unor surse de curent comandate în tensiune. Tensiunile corespunzatoare portilor u si v au fost notate, pentru celula i, cu u_i si v_i. Fiecare celula este conectata cu celulele vecine din stânga si dreapta.

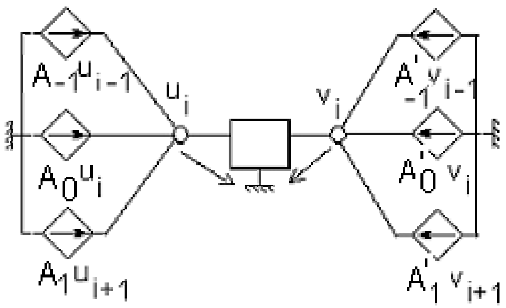


Fig 36: Interconexiunile în CNN

Parametrii A_{-1}, A_0, A_1 si respectiv A'_{-1}, A'_0, A'_1 template-uri. Se presupune pentru generalitate ca cele doua straturi au template-uri diferite. Un caz particular este acela în care $A_{-1}= A'_{-1}$ si $A_1=A'_1$ (template-uri identice pe cele doua straturi).

Template-uri asimetrice si template-uri simetrice

Expresia operatorului de interconexiune este:

$$O_{1D}(u_i) = A_{-1}u_{i-1} + A_0 u_{i+1} + A_1 u_i \qquad (7.1.)$$

Dupa cum s-a specificat si mai sus, template-urile corespunzatoare celor doua straturi sunt în general diferite. Este simetric un template pentru care parametrii A_{-1} si A_1 sunt egali.

În cele ce urmeaza se prezinta câteva tipuri de conditii la limita (de granita). Se va studia un CNN cu M celule.

Conditii de granita

Conditiile de granita se refera la modul în care se conecteaza celulele de la extremitatile CNN-ului [66,69,73,74,85]. Aceste celule au indecsii –1 si respectiv M.

Cel mai general caz este acela în care nu exista celule de granita, adica acel caz în care se studiaza un CNN infinit. În cazul particular în care se vorbeste de semnale periodice (analoage regimului permanent din cazul semnalelor temporale) CNN-ul infinit va fi înlocuit cu un CNN care prezinta asa-zisele conditii de granita periodice. Acest lucru implica faptul ca lungimile de unda ale armonicilor sunt submultiplii ai dimensiunii CNN.

Este posibila co-existenta mai multor tipuri de conditii de granita pe straturile u si respectiv v.

Conditiile de granita sunt descrise prin intermediul seturilor de ecuatii:

- *conditii de granita de tip periodic (inel)*

$$u(-1,t) = u(M-1,t), \quad u(0,t) = u(M,t) \tag{7.2.}$$

- *conditii de granita de tip zero-flux*

$$u(-1,t) = u(0,t), \quad u(M,t) = u(M-1,t) \tag{7.3.}$$

- *conditii de granita de tip anti-zero-flux*

$$u(M,t) = -u(M-1,t), \quad u(-1,t) = -u(0,t) \tag{7.4.}$$

- *conditii de granita de tip zero*

$$u(-1,t) = 0, \quad u(M,t) = 0 \tag{7.5.}$$

- *conditii de granita de tip quasi-zero flux*

$$u(-1,t) = u(1,t), \quad u(M,t) = u(M-2,t) \tag{7.6.}$$

Ecuatii si solutie

Forma generala a ecuatiilor

Forma generala a ecuatiilor care descriu functionarea sistemului format din celule de ordinul 2, dublu cuplate este [60-62, 81-87]:

$$\begin{cases} \dfrac{du_i(t)}{dt} = F(u_i, v_i) + D_u O_{1D}(u_i) & i = 0..M-1 \\ \dfrac{dv_i(t)}{dt} = G(u_i, v_i) + D_v O_{1D}(v_i) \end{cases} \tag{7.7.}$$

S-au introdus fata de ecuatiile (6.1) termenii corespunzatori legaturilor celulei ce are indicele i cu celulele vecine (influenta operatorului O_{1D}).
Pentru studiul stabilitatii sistemului, acesta se liniarizeaza în jurul unui punct de echilibru (uzual originea). Se obtine în acest fel urmatorul sistem format din 2M ecuatii:

$$\begin{cases} \dfrac{du_i(t)}{dt} = g\,(f_u u_i + f_v v_i) + D_u O_{1D}(u_i) & i = 0..M-1 \\ \dfrac{dv_i(t)}{dt} = g\,(g_u u_i + g_v v_i) + D_v O_{1D}(v_i) \end{cases} \tag{7.8.}$$

în care parametrii f_u, f_v, g_u, g_v sunt elementele Jacobianului corespunzator sistemului liniarizat.

Rezolvarea ecuatiilor

Tehnica de decuplare

Solutionarea sistemului consta în utilizarea unei schimbari de variabila [60-62]:

$$\begin{cases} u_i = \sum_{m=0}^{M-1} \Phi_M(m,i)\, \hat{u}_m & i = 0..M-1 \\ v_i = \sum_{m=0}^{M-1} \Phi_M(m,i)\, \hat{v}_m \end{cases} \tag{7.9.}$$

unde *F $_M$(m,i)* sunt functii spatiale ortogonale în raport cu produsul scalar.
Relatiile de inversiune sunt:

$$\begin{cases} \hat{u}_m = \sum_{m=0}^{M-1} \Phi^*{}_M(m,i) u_i & i = 0..M-1 \\ \hat{v}_m = \sum_{m=0}^{M-1} \Phi^*{}_M(m,i) v_i \end{cases} \tag{7.10.}$$

În cazul conditiilor la limita periodice functiile
Cel mai general caz cu care se va lucra este acela în care functiile *F $_M$(m,i)* sunt de forma [81-87]:

$$\Phi_M(w(m), j(m), i) = e^{j(w(m)i + j(m))} \qquad (7.11.)$$

Acestea sunt functii proprii ale operatorului O_{1D}:

$$O_{1D}(\Phi_M(w(m), j(m), i)) = K_{1D}(m)\Phi_M(w(m), j(m), i) \qquad (7.12.)$$

Înlocuind în sistemul de ecuatii (7.8) relatiile (7.9) si (7.12) si facând produs scalar se obtine

$$\begin{bmatrix}\dot{\hat{u}}_m(t)\\ \dot{\hat{v}}_m(t)\end{bmatrix} = \left(g\begin{bmatrix} f_u & f_v\\ g_u & g_v\end{bmatrix} + \begin{bmatrix} K_{1D}(m)D_u & 0\\ 0 & K_{1D}'(m)D_v\end{bmatrix}\right)\begin{bmatrix}\hat{u}_m(t)\\ \hat{v}_m(t)\end{bmatrix} + g\begin{bmatrix}\hat{e}_m'\\ \hat{e}_m\end{bmatrix} \qquad (7.13.)$$

S-a luat în discutie cazul general în care caracteristica elementului neliniar nu trece prin origine (caz care se poate asemana unuia în care sistemul este excitat cu un semnal spatial constant în timp de o forma particulara) si când template-urile pe cele doua straturi sunt diferite. În relatia (7.13) *$K_{1D}(m)$* si *$K_{1D}'(m)$* sunt valorile proprii spatiale corespunzatoare stratului notat cu u si respectiv stratului v. Expresia acestor valori proprii va fi dedusa ulterior.
Parametrul $\varepsilon_m^{\wedge}$ reprezinta valoarea componentei spectrale cu indexul m a semnalului spatial constituit de valorile sursei de curent continuu ale fiecarei celule (deplasamentul caracteristicii liniare pe portiuni a elementului neliniar, pe verticala fata de origine).
Se observa ca forma sistemului de ecuatii este una clasica, în care se pot izola matricea de tranzitie a starilor si matricea asociata excitatiei:

$$A = g\begin{bmatrix} f_u & f_v\\ g_u & g_v\end{bmatrix} + \begin{bmatrix} K_{1D}(m)D_u & 0\\ 0 & K_{1D}'(m)D_v\end{bmatrix} \qquad B = g \qquad (7.14.)$$

Semnalul de intrare asociat cu sursele de curent constant dar cu valoare neomogena poate fi vazut ca o forma particulara de excitatie:

$$\dot{X} = AX + Be \qquad Y = X \qquad (7.15.)$$

Solutia sistemului liniarizat în jurul originii este data în relatia:

$$\begin{cases} \hat{u}_m(t) = a_m(\hat{e}_m, \hat{e}_m', \hat{u}_m(0), \hat{v}_m(0))e^{l_{m_1}t} + \\ \qquad\qquad b_m(\hat{e}_m, \hat{e}_m', \hat{u}_m(0), \hat{v}_m(0))e^{l_{m_2}t} + f_1(\hat{e}_m, \hat{e}_m') \\ \hat{v}_m(t) = c_m(\hat{e}_m, \hat{e}_m', \hat{u}_m(0), \hat{v}_m(0))e^{l_{m_1}t} + \\ \qquad\qquad d_m(\hat{e}_m, \hat{e}_m', \hat{u}_m(0), \hat{v}_m(0))e^{l_{m_2}t} + f_2(\hat{e}_m, \hat{e}_m') \end{cases} \qquad (7.16.)$$

Parametrii λ_{m1} si λ_{m2} sunt solutiile ecuatiei caracteristice:

$$\begin{aligned} l_m^2 - l_m[g(f_u + g_v) + D_u K_{1D}(m) + D_v K_{1D}^{'}(m)] + D_u D_v K_{1D}(m) K_{1D}^{'}(m) \\ + g(D_v K_{1D}^{'}(m) f_u + D_u K_{1D}(m) g_v) + g^2 (f_u g_v - f_v g_u) = 0 \end{aligned} \tag{7.17.}$$

iar f_1 si f_2 reprezinta:

$$\begin{cases} f_1(\hat{e}_m, \hat{e}_m^{'}) = \dfrac{-g(gg_v + K_{1D}^{'} D_v)\hat{e}_m + g^2 f_v \hat{e}_m^{'}}{(gf_u + K_{1D} D_u)(gg_v + K_{1D}^{'} D_v) - g^2 f_v g_u} \\ f_2(\hat{e}_m, \hat{e}_m^{'}) = \dfrac{g^2 g_u \hat{e}_m - g(gf_u + K_{1D} D_u)\hat{e}_m^{'}}{(gf_u + K_{1D} D_u)(gg_v + K_{1D}^{'} D_v) - g^2 f_v g_u} \end{cases} \tag{7.18.}$$

Cu notatiile:

$$p_m = \frac{l_{m1} - gf_u - D_u K_{1D}}{gf_v}; \; q_m = \frac{l_{m2} - gf_u - D_v K_{1D}^{'}}{gg_u} \tag{7.19.}$$

constantele de integrare a_m, b_m, c_m si d_m au expresiile:

$$\begin{aligned} & a_m(\hat{e}_m, \hat{e}_m^{'}, \hat{u}_m(0), \hat{v}_m(0)) = \\ & \qquad -\frac{1}{p_m q_m - 1}\hat{u}_m(0) + \frac{q_m}{p_m q_m - 1}\hat{v}_m(0) + \frac{f_1(\hat{e}_m, \hat{e}_m^{'}) - q_m f_2(\hat{e}_m, \hat{e}_m^{'})}{p_m q_m - 1} \\ & d_m(\hat{e}_m, \hat{e}_m^{'}, \hat{u}_m(0), \hat{v}_m(0)) = \\ & \qquad \frac{p_m}{p_m q_m - 1}\hat{u}_m(0) - \frac{1}{p_m q_m - 1}\hat{v}_m(0) + \frac{f_2(\hat{e}_m, \hat{e}_m^{'}) - p_m f_1(\hat{e}_m, \hat{e}_m^{'})}{p_m q_m - 1} \\ & b_m(\hat{e}_m, \hat{e}_m^{'}, \hat{u}_m(0), \hat{v}_m(0)) = \\ & \qquad \frac{p_m q_m}{p_m q_m - 1}\hat{u}_m(0) - \frac{q_m}{p_m q_m - 1}\hat{v}_m(0) + \frac{f_2(\hat{e}_m, \hat{e}_m^{'}) - p_m f_1(\hat{e}_m, \hat{e}_m^{'})}{p_m q_m - 1} q_m \\ & c_m(\hat{e}_m, \hat{e}_m^{'}, \hat{u}_m(0), \hat{v}_m(0)) = \\ & \qquad -\frac{p_m}{p_m q_m - 1}\hat{u}_m(0) + \frac{p_m q_m}{p_m q_m - 1}\hat{v}_m(0) + \frac{f_1(\hat{e}_m, \hat{e}_m^{'}) - q_m f_2(\hat{e}_m, \hat{e}_m^{'})}{p_m q_m - 1} p_m \end{aligned} \tag{7.20.}$$

Solutia în domeniu timp pentru sistemul de ecuatii liniarizat este:

$$\begin{aligned} u_i(t) &= \sum_{m=0}^{M-1} \left[\begin{aligned} & a_m(\hat{e}_m, \hat{e}_m^{'}, \hat{u}_m(0), \hat{v}_m(0)) e^{l_{m_1} t} + \\ & \qquad b_m(\hat{e}_m, \hat{e}_m^{'}, \hat{u}_m(0), \hat{v}_m(0)) e^{l_{m_2} t} + f_1(\hat{e}_m, \hat{e}_m^{'}) \end{aligned} \right] \Phi_M(m, i) \\ v_i(t) &= \sum_{m=0}^{M-1} \left[\begin{aligned} & c_m(\hat{e}_m, \hat{e}_m^{'}, \hat{u}_m(0), \hat{v}_m(0)) e^{l_{m_1} t} + \\ & \qquad d_m(\hat{e}_m, \hat{e}_m^{'}, \hat{u}_m(0), \hat{v}_m(0)) e^{l_{m_2} t} + f_2(\hat{e}_m, \hat{e}_m^{'}) \end{aligned} \right] \Phi_M(m, i) \end{aligned} \tag{7.21.}$$

Din forma generala a solutiei, se poate vedea ca depinde de conditiile initiale, valorile curentilor debitati de sursele independente de curent, valorile proprii temporale si de forma functiilor proprii, care este legata de conditiile de granita, dupa cum se va vedea în continuare.

Functii proprii si valori proprii pentru exponentiale (conditii de frontiera periodice si template-uri asimetrice)

Dupa cum am subliniat în paragraful precedent, cel mai general tip de functii proprii considerat este cel corespunzator bazei transformatei Fourier discrete.
Template-ul dat mai jos descrie în general operatorul asimetric O_{1D}:

A_{-1}	A_0	A_1

Functia reprezentata în relatia (7.11) este functie proprie pentru operatorul O_{1D}, valoarea proprie fiind:

$$K_{1D}(m) = A_0 + (A_{-1} + A_1)\cos(w(m)) + j(A_1 - A_{-1})\sin(w(m)) \quad (7.22.)$$

Pentru cazul general valorile proprii spatiale sunt numere complexe. Pentru template-uri simetrice ($A_{-1}=A_1$):

$$K_{1D} = A_0 + 2A_1\cos(w(m)) \quad (7.23.)$$

valorile proprii spatiale sunt reale

Demonstratie:

$$\begin{aligned}
O(\Phi_M(w(m), j(m), i)) &= A_{-1}\Phi_M(w(m), j(m), i-1) + A_0\Phi_M(w(m), j(m), i) \\
&\quad + A_1\Phi_M(w(m), j(m), i+1) \\
&= \{A_0 + (A_1 + A_{-1})\cos(w(m)) + (A_1 - A_{-1})\sin(w(m))\}\Phi_M(w(m), j(m), i) \\
&= K_{1D}(A_{-1}, A_0, A_1, w(m))\Phi_M(w(m), j(m), i)
\end{aligned}$$

În cazul în care $A_{-1}=A_1$, partea imaginara devine nula si se obtine (7.23).
Din demonstratie rezulta faptul ca valorile proprii spatiale depind de parametrii template-ului si de modul (frecventa spatiala) la care sunt atasati.

Forma generala a solutiei pentru cazul functiilor proprii exponentiale

Forma generala a solutiei pentru cazul functiilor proprii exponentiale este data în relatia de mai jos:

$$u_i(t) = \sum_{m=0}^{M-1} \begin{bmatrix} a_m(\hat{e}_m, \hat{e}_m^{'}, \hat{u}_m(0), \hat{v}_m(0))e^{l_{m_1} t} + \\ b_m(\hat{e}_m, \hat{e}_m^{'}, \hat{u}_m(0), \hat{v}_m(0))e^{l_{m_2} t} + f_1(\hat{e}_m, \hat{e}_m^{'}) \end{bmatrix} e^{j(w(m)i + j(m))}$$

$$v_i(t) = \sum_{m=0}^{M-1} \begin{bmatrix} c_m(\hat{e}_m, \hat{e}_m^{'}, \hat{u}_m(0), \hat{v}_m(0))e^{l_{m_1} t} + \\ d_m(\hat{e}_m, \hat{e}_m^{'}, \hat{u}_m(0), \hat{v}_m(0))e^{l_{m_2} t} + f_2(\hat{e}_m, \hat{e}_m^{'}) \end{bmatrix} e^{j(w(m)i + j(m))} \tag{7.24.}$$

Conditiile initiale

Conditiile initiale joaca un rol cheie în formarea pattern-urilor [72, 86]. S-au utilizat conditii initiale mici care asigura un oarecare timp de evolutie pâna la atingerea primei zone cu alta panta a caracteristicii liniare pe portiuni (în cazul cel mai general când avem de a face cu doua elemente neliniare în celula).

Utilizând relatiile de mai sus se pot controla în cel mai strict mod solutiile sistemului de ecuatii si deci pattern-ul la momentul t_x.

Suprafata[1] de dispersie pentru cazul exponentialelor

Conditia necesara si suficienta pentru ca sistemul sa devina instabil este ca acesta sa aiba cel putin o componenta spectrala în semnalul constituit de conditiile initiale cu partea reala a valorii proprii temporale corespunzatoare pozitiva:

$$\mathrm{Re}(l_{1,2}(K_{1D}(m), K_{1D}^{'}(m))) = \mathrm{Re}\{g\frac{f_u + g_v}{2} + \frac{D_u K_{1D}(m) + D_v K_{1D}^{'}(m)}{2} \pm \sqrt{\left[g\frac{(g_v - f_u)}{2} - \frac{(D_u K_{1D}(m) - D_v K_{1D}^{'}(m))}{2}\right]^2 + g^2 f_v g_u}\} > 0 \tag{7.25.}$$

Relatia de mai sus reprezinta expresia radacinilor ecuatiei caracteristice în cazul când fiecarui strat îi corespunde un template.

În general, $K_{1D}(m)$ si $K_{1D}^{'}(m)$ au valori complexe. În consecinta, radacinile ecuatiei caracteristice pot fi complexe, dar nu în mod necesar complex conjugate. Stabilitatea în legatura cu un anumit mod este decisa de valoarea partii reale a valorii proprii temporale (pozitiva sau negativa), în timp ce evolutia unui anumit mod (faptul ca se face într-o maniera oscilanta sau nu) este decisa de partea imaginara a valorii proprii temporale corespunzatoare.

[1] Reprezinta valorile proprii temporale ale sistemului în functie de valorile proprii spatiale

În Fig. 37 este reprezentata suprafata de dispersie pentru cazul restrictiv în care template-ul este simetric (valori proprii spatiale reale):

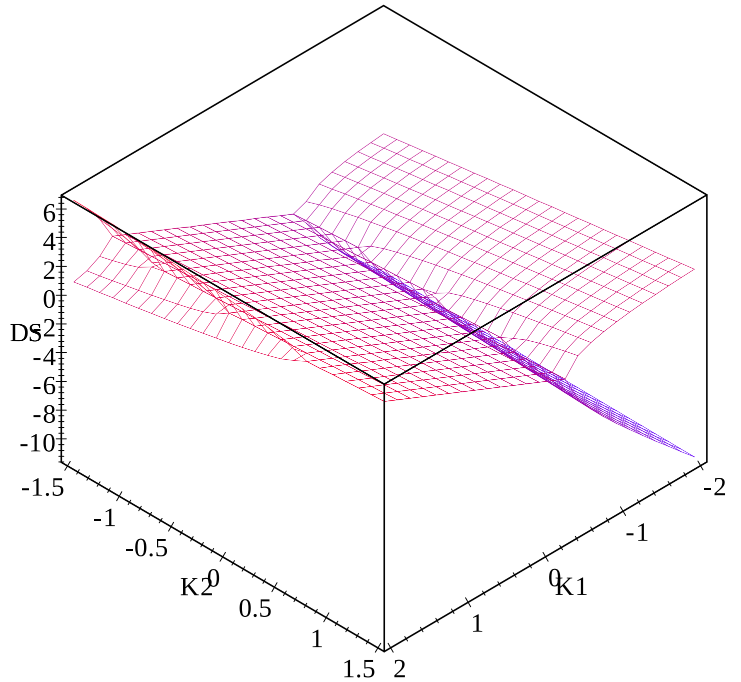


Fig. 37: Partea reala a ambelor valori proprii temporale în functie de valorile proprii spatiale pentru fiecare strat

În cazul în care ambele straturi au acelasi template, se obtine o *curba* de dispersie si nu o suprafata, care este reprezentata mai jos:

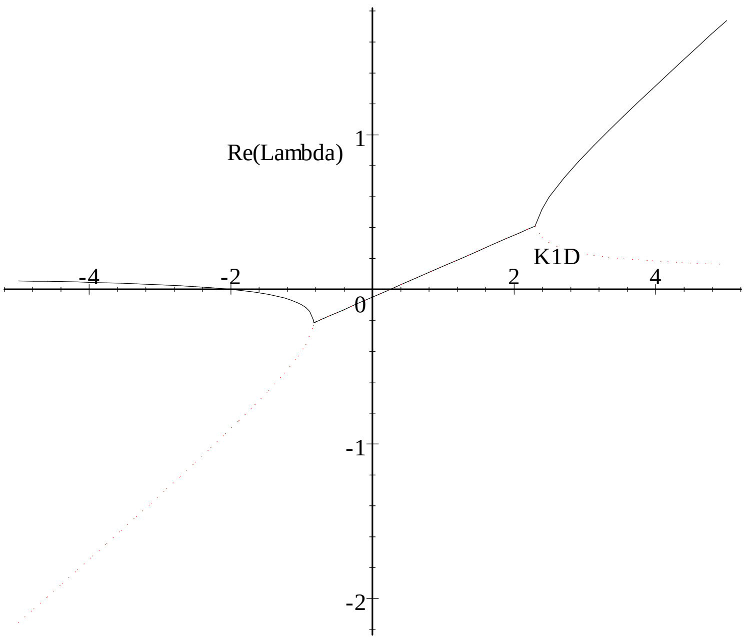


Fig. 38: Partea reala a ambelor valori proprii temporale în functie de valoarea proprie spatiala (aceeasi pentru fiecare strat)

S-a reprezentat cu linie solida radacina cu semnul + si cu linie punctata radacina cu semn -. Sa observam ca exista o regiune din curba de dispersie pentru care radacinile sunt complex conjugate (cazul cu template-uri simetrice pentru fiecare din cele doua straturi) si doua regiuni în fiecare din cele doua parti în care radacinile sunt reale si distincte. Aceasta înseamna ca pattern-ul va creste "din doua motive" în aceste regiuni laterale.
Mai mult, putem obtine din jocul parametrilor aceeasi curba plasata în regiuni diferite. Regiunea centrala (legata de valorile proprii temporale complex conjugate) poate avea panta negativa numai prin simpla schimbare a semnului unui parametru (D_v).
O conditie necesara pentru ca zona din mijloc a curbei de dispersie sa existe este ca produsul $f_v g_u$ sa fie negativ [81,82]. În toate cazurile în care produsul $f_v g_u$ este pozitiv valorile proprii temporale sunt reale.

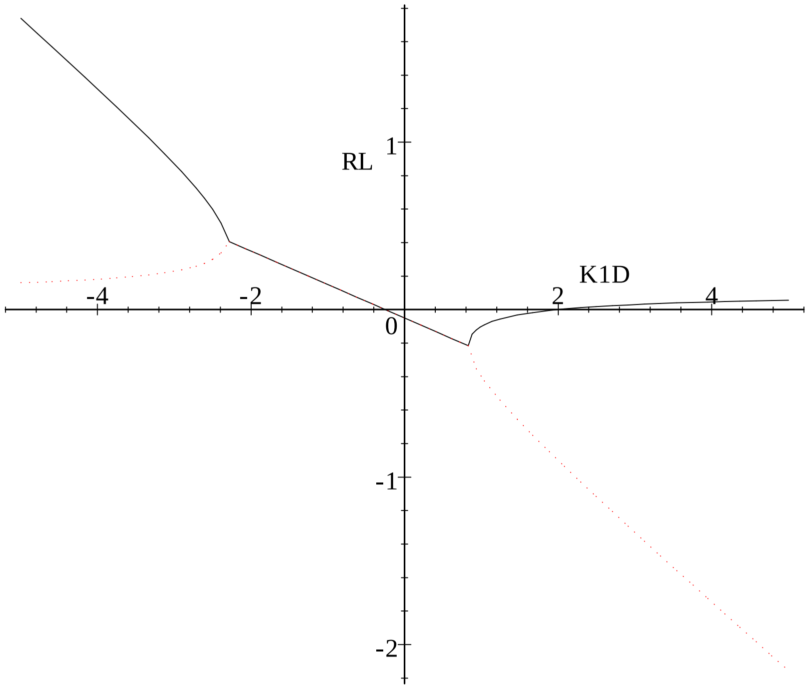


Fig. 39: Partea reala a ambelor valori proprii temporale în functie de valoarea proprie spatiala (aceeasi pentru ambele straturi) când se schimba semnul lui D_v (D_v >>D_u)

Decuplarea straturilor

În cazul în care termenul $\gamma^2 f_v g_u$ este mic (sau zero) acesta se poate neglija si se obtine un sistem format din doua CNN-uri cu celule de ordinul 1, care functioneaza independent una fata de cealalta. Radacinile ecuatiei caracteristice sunt date mai jos:

$$l_{m1}(m) \cong gf_u + D_u K_{1D}(m)$$
$$l_{m2}(m) \cong gg_v + D_v K_{1D}'(m) \qquad (7.26.)$$

Prima radacina depinde numai de parametrii asociati primului strat (asociat cu indexul u), în timp ce a doua radacina depinde numai de cei corespunzatori sistemului v (asociat cu indexul v). Curba de dispersie corespunzând fiecarui sub-sistem separat format din celule de ordinul I este o dreapta cu panta depinzând de semnul lui D_u si respectiv D_v. Aceasta corespunde rezultatelor care s-au raportat pâna în acest moment pentru CNN-uri cu celule de ordinul I.

Template-uri simetrice

În ceea ce urmeaza ne vom îndrepta atentia spre template-urile simetrice. Dupa cum am mai mentionat, pentru aceasta familie de template-uri decuplarea ecuatiilor este posibila si pentru alte tipuri de conditii de granita si în consecinta functii proprii. Functiile proprii sunt în toate aceste cazuri reale si tot reale vor fi si valorile proprii spatiale.

Dupa cum s-a aratat într-unul din paragrafele precedente, functia exponentiala "trece" nedeformata prin operatorul spatial liniar O_{1D}. Vom trata în mod global aceasta familie de functii proprii tinând cont de o forma care sa includa toate functiile proprii cunoscute în acest moment [69]:

Forma template-ului luat în discutie este dat mai jos:

A_{-1}	A_0	A_1

Valoarea proprie asociata functiei proprii de tip exponentiala este aceeasi si pentru functia proprie data mai jos:

$$\Phi_M(-w(m),-j(m),i) = e^{-j(w(m)i+j(m))} \qquad (7.27.)$$

Demonstratie:

Rezultatul este evident (folosind 7.23) din caracterul par al functiei cos si din faptul ca parametrul φ(m) nu apare în expresia valorii proprii spatiale.

Functiile:

$$\begin{aligned} \Phi_M'(w(m), j(m), i) &= \cos(w(m)i + j(m)) \\ \Phi_M''(w(m), j(m), i) &= \sin(w(m)i + j(m)) \end{aligned} \qquad (7.28.)$$

sunt functii proprii pentru operatorul O_{1D} si corespunzator au aceeasi valoare proprie spatiala ca si functia exponentiala pentru cazul template-ului simetric (7.23):

Demonstratie:
Sa demonstram prima afirmatie:

$$O_{1D}(\Phi_M{}^{'}(w(m), j(m), i)) =$$
$$\frac{1}{2}\left[O_{1D}(\Phi_M(w(m), j(m), i)) + O_{1D}(\Phi_M(-w(m), -j(m), i))\right] =$$
$$\frac{1}{2}\left[K_{1D}\Phi_M(w(m), j(m), i) + K_{1D}\Phi_M(-w(m), -j(m), i)\right] =$$
$$K_{1D}\Phi_M{}^{'}(w(m), j(m), i)$$

A doua afirmatie poate fi demonstrata în mod similar:

$$O_{1D}(\Phi_M{}^{''}(w(m), j(m), i)) =$$
$$\frac{1}{2j}\left[O_{1D}(\Phi_M(w(m), j(m), i)) - O_{1D}(\Phi_M(-w(m), -j(m), i))\right] =$$
$$\frac{1}{2}\left[K_{1D}\Phi_M(w(m), j(m), i) - K_{1D}\Phi_M(-w(m), -j(m), i)\right] =$$
$$K_{1D}\Phi_M{}^{''}(w(m), j(m), i)$$

Se mai poate face observatia conform careia a doua functie proprie putea fi pusa în forma primeia daca se facea schimbarea de variabila urmatoare:

$$j(m) \rightarrow j(m) - \frac{p}{2} \qquad (7.29.)$$

Conditii de granita speciale

Conditiile de granita despre care s-a discutat în sectiunea *Conditii de granita* corespund fiecare unui anumit tip de functie proprie. Conditiile de granita pot fi *pure* si respectiv *impure*. Denumim conditii de granita *pure* cazul când se considera acelasi tip de conditii de granita la fiecare capat al retelei celulare 1D. Pentru conditii de granita diferite pentru fiecare capat al retelei, acestea sunt impure.

În continuare se prezinta functiile proprii pentru ambele tipuri de conditii de granita (pure si impure) Utilizând forma generala a functiilor proprii si particularizând aceasta forma generala pentru fiecare tip de conditii de granita (fie pure ori impure) si în consecinta pentru fiecare din functiile proprii corespunzatoare conditiilor de granita se obtine tabelul 1 [69]:

Tipul conditiei de granita	$\Phi_M(w(m), j(m), i)$	$w(m)$	$j(m)$	K_{1D}
periodic	$e^{j\frac{2p}{M}mi}$	$\frac{2p}{M}m$	0	$A_0 + 2A_1 \cos\frac{2p}{M}m$
zf-zf	$\cos\frac{(2i+1)}{2M}mp$	$\frac{p}{M}m$	$\frac{p}{2M}m$	$A_0 + 2A_1 \cos\frac{p}{M}m$
azf-azf	$\sin\frac{(2i+1)(m+1)p}{2M}$	$\frac{p}{M}(m+1)$	$\frac{p}{2M}(m+1)\left\{-\frac{p}{2}\right\}$	$A_0 + 2A_1 \cos\frac{p}{M}(m+1)$
z-z	$\sin\frac{(i+1)(m+1)p}{M+1}$	$\frac{p}{M+1}(m+1)$	$\frac{p}{M+1}(m+1)\left\{-\frac{p}{2}\right\}$	$A_0 + 2A_1 \cos\frac{p}{M+1}(m+1)$
qzf-qzf	$\cos\frac{mip}{M-1}$	$\frac{p}{M-1}m$	0	$A_0 + 2A_1 \cos\frac{p}{M-1}m$
azf-zf	$\sin\frac{(2i+1)(2m+1)p}{4M}$	$\frac{p}{2M}(2m+1)$	$\frac{p}{4M}(2m+1)\left\{-\frac{p}{2}\right\}$	$A_0 + 2A_1 \cos\frac{p}{2M}(2m+1)$
z-zf	$\sin\frac{(i+1)(2m+1)p}{2M+1}$	$\frac{p}{2M+1}(2m+1)$	$\frac{p}{2M+1}(2m+1)\left\{-\frac{p}{2}\right\}$	$A_0 + 2A_1 \cos\frac{p}{2M+1}(2m+1)$

(continuare)

Tipul conditiei de granita	$\Phi_M(w(m), j(m), i)$	$w(m)$	$j(m)$	K_{1D}
z-qzf	$\sin\frac{(i+1)(2m+1)p}{2M}$	$\frac{p}{2M}(2m+1)$	$\frac{p}{2M}(2m+1)\left\{-\frac{p}{2}\right\}$	$A_0 + 2A_1\cos\frac{p}{2M}(2m+1)$
azf-qzf	$\sin\frac{(2i+1)(2m+1)p}{2(2M-1)}$	$\frac{p}{2M-1}(2m+1)$	$\frac{p}{2(2M-1)}(2m+1)\left\{-\frac{p}{2}\right\}$	$A_0 + 2A_1\cos\frac{p}{2M-1}(2m+1)$
z-azf	$\sin\frac{2(i+1)(m+1)p}{2M+1}$	$\frac{2p}{2M+1}(m+1)$	$\frac{2p}{2M+1}(m+1)$	$A_0 + 2A_1\cos\frac{2p}{2M+1}(m+1)$

Tabelul 1: Functii si valori proprii pentru diferite tipuri de conditii de granita (1D)

Curba de dispersie vazuta ca functie de moduri

Curba de dispersie poate fi interpretata ca reprezentarea caracteristicii de amplitudine a unui filtru spatial în raport cu conditiile initiale, în sensul ca, pornind de la conditii initiale mici filtrul "amplifica" modurile care au partea reala a valorii proprii temporale corespunzatoare pozitiva. În consecinta, reprezentarea partii reale a valorilor proprii temporale în functie de moduri poate fi facuta pe baza proprietatii de monotonie în raport cu m a tuturor functiilor proprii studiate, pe tot intervalul în care m variaza de la 0 la M-1. În continuare se considera cazul template-urilor simetrice identice pe ambele straturi.

Expresia matematica a valorii proprii spatiale $K_{1D}(m)$ arata ca aceasta apartine intervalului [min($A_0 \pm 2A_1$), max($A_0 \pm 2A_1$)]. Lucrurile pot fi privite intuitiv în felul urmator: se alege o fereastra în cadrul curbei de dispersie, a carei pozitie este data de parametrul A_0 al template-ului si a carei largime este data de parametrul A_1 al aceluiasi template. Pentru a pastra tipul de monotonie al curbei de dispersie în functie de valorile proprii spatiale la schimbarea de variabila, trebuie sa alegem semnul lui A_1 negativ (filtrele trece jos sa ramâna trece jos si în coordonate functie de moduri, etc). Astfel cu aceasta conventie $K_{1D}(m)$ va apartine intervalului [A_0-2A_1, A_0+2A_1] cu A_0 si A_1 pozitivi.

Pentru prima curba de dispersie reprezentata vom izola o fereastra corespunzatoare parametrilor $A_0=2$ si $A_1=-1$. Intervalul în care $K_{1D}(m)$ ia valori este [0,4]. În consecinta, curba de dispersie reprezentata ca functie de moduri va fi distorsionata (dependenta neliniara, dar monotona a lui K_{1D} de m), dar va corespunde unei ferestre corespunzatoare valorilor lui $K_{1D}(m)$ între 0 si 4:

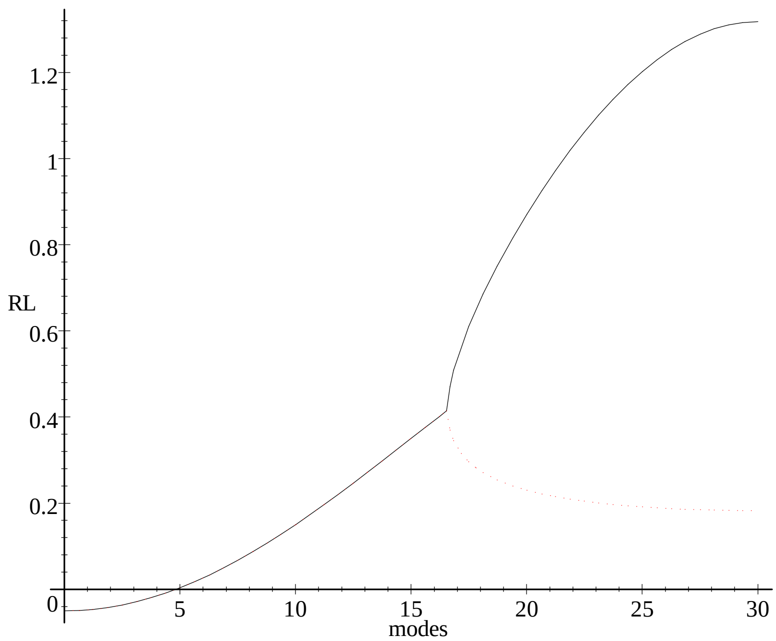


Fig. 40: Partea reala a ambelor valori proprii temporale functie de moduri (aceeasi valoare pentru ambele straturi)

Linia solida (cea de deasupra) reprezinta locul geometric al partii reale a valorii proprii temporale cu semnul + când m parcurge intervalul [0,29] (30 celule) în timp ce linia punctata reprezinta acelasi lucru pentru solutia cu semnul -.
De exemplu, luând în considerare chiar figura de mai sus, modurile cu indecsi mai mici de 17 se vor dezvolta într-o maniera oscilanta crescator exponential, pe când cele cu indecsi mai mari decât 17 se vor dezvolta numai într-un mod exponential (nu vor avea caracter oscilant).
Utilizând aceasta tehnica se poate imagina usor un mod de a proiecta filtre spatiale de acest tip având aceleasi template-uri simetrice pe ambele straturi:

- se seteaza valorile pentru toti parametrii tinând cont de punctele esentiale de pe curba de dispersie cu exceptia lui A_0 si al lui A_1. În aceasta etapa, forma curbei de dispersie ca functie de K_{1D} este cunoscuta.
- se seteaza valoarea pentru parametrul A_0 (pentru plasarea ferestrei de interes) si apoi se seteaza valoarea parametrului A_1 pentru a se stabili largimea ferestrei (cât din curba de dispersie se va lua în considerare).

Utilizând aceasta tehnica se pot obtine toate tipurile de filtre spatiale TJ, TS, TB).

Cazuri particulare

Pattern-uri Turing

Cazul „pattern-uri Turing” corespunde unui template simetric, acelasi pentru ambele straturi, de forma:

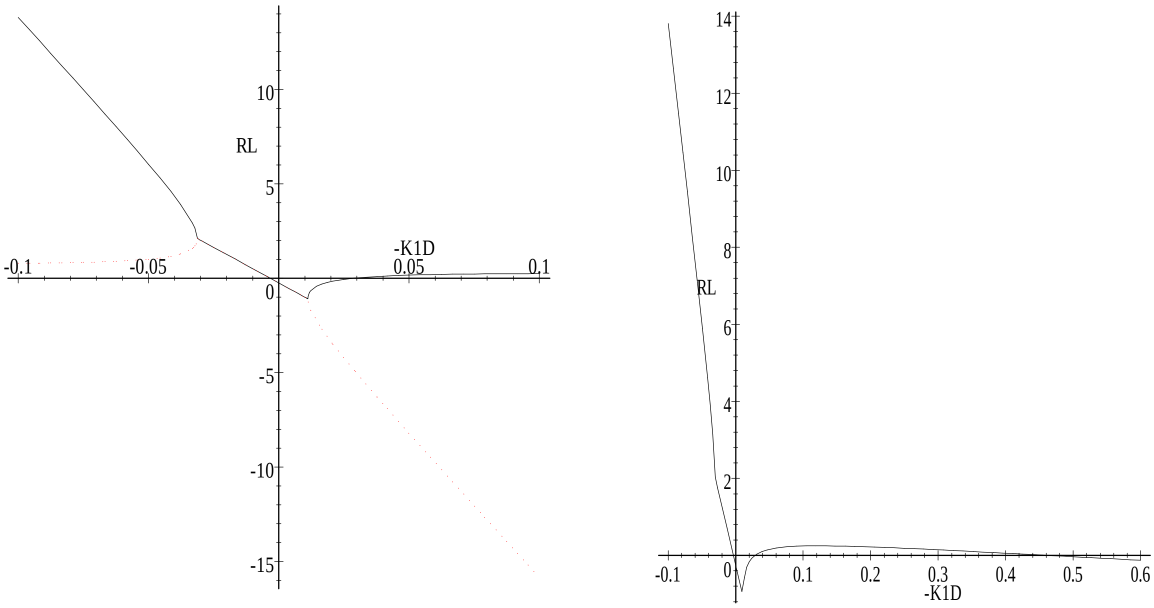
-1	2	-1

Pentru a se obtine pattern-uri Turing, trebuie sa fie satisfacute urmatoarele conditii [60-62]:

$$\begin{aligned} & f_u + g_v < 0 \\ & f_u g_v - f_v g_u > 0 \\ & D_v f_u + D_u g_v > 0 \\ & (D_v f_u - D_u g_v)^2 + 4 D_u D_v f_v g_u > 0 \end{aligned} \qquad (7.30.)$$

Un set de parametrii pentru care un asemenea sistem poate produce pattern-uri Turing este: γ=5, f_u=0.1, f_v=-1, g_u=0.1, g_v=-0.2, D_u=1, D_v=150. (M=30).

Curba de dispersie corespunzatoare este reprezentata în figurile de mai jos:



a) ambele solutii b) radacina cea mai mare

Fig. 41: Partea reala a valorilor proprii temporale functie de valoarea proprie spatiala (aceeasi pentru ambele straturi)

Se reaminteste ca s-a facut reprezentarea în functie de $-K_{1D}$ din cauza conventiei diferite în cazul pattern-urilor Turing. Cu notatia adoptata în aceasta lucrare, în cazul pattern-urilor Turing valorile proprii spatiale au tot timpul semnul negativ [61].

Modalitati de control ale curbei de dispersie

În paragraful "Curba de dispersie vazuta ca functie de moduri" s-a expus principiul calitativ care conduce proiectarea unui filtru liniar spatial. În acest paragraf se vor da câteva relatii cantitative si se va da un exemplu de proiectare a unui asemenea filtru.

Puncte semnificative ale curbei de dispersie

Dupa cum se poate vedea în reprezentarile acesteia, curba de dispersie cuprinde în general trei regiuni: 2 laterale în care valorile proprii sunt reale si distincte si una centrala în care valorile proprii sunt complex conjugate în cazul template-ului simetric si acelasi pe ambele straturi.

În particular, zona valorilor proprii complexe se poate restrânge până la disparitie, ca în figura de mai jos, caz în care $f_v g_u > 0$:

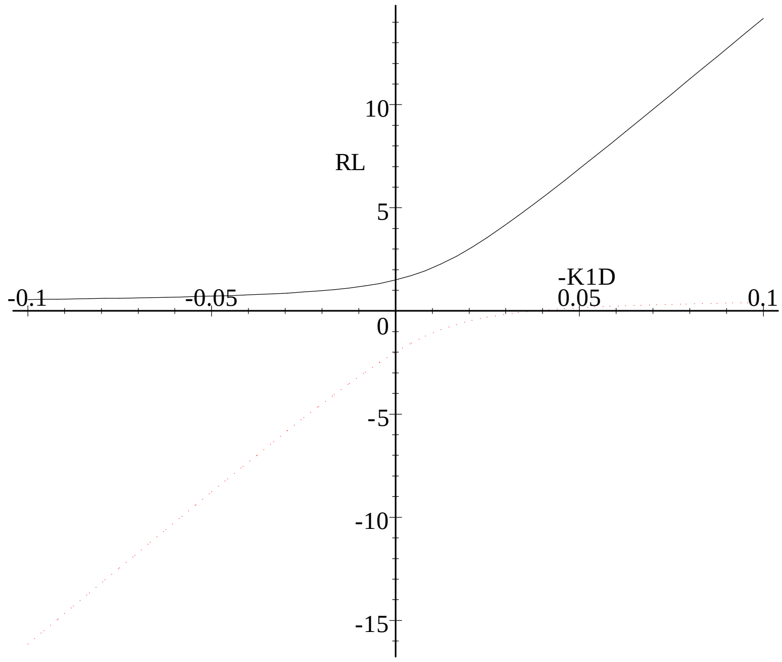


Fig. 42: Partea reala a valorilor proprii temporale functie de valoarea proprie spatiala (aceeasi pentru ambele straturi) pentru $f_v g_u > 0$

Zona de valori proprii complex conjugate este delimitata de urmatoarele doua valori:

$$K_{1Dleft} = \min\left\{-\frac{g}{D_u - D_v}\left(f_u - g_v \pm 2\sqrt{-f_v g_u}\right)\right\}$$
$$K_{1Dright} = \max\left\{-\frac{g}{D_u - D_v}\left(f_u - g_v \pm 2\sqrt{-f_v g_u}\right)\right\} \quad (7.31.)$$

Largimea zonei de mijloc este data de relatia:

$$\Delta K_{1D} = -\frac{4g\sqrt{-f_v g_u}}{D_u - D_v} \quad (7.32.)$$

Coordonata mijlocului zonei de mijloc este data de relatia:

$$K_{1Dmiddle} = -\frac{g}{2(D_u - D_v)}\left(f_u - g_v\right) \quad (7.33.)$$

Din relatiile de mai sus [81,84] se vede înca o data ca o conditie necesara pentru ca zona de mijloc sa existe (numerele K_{1Dleft} si $K_{1Dright}$ sa fie reale) trebuie ca produsul $f_v g_u$ sa fie negativ. La limita daca $f_v g_u = 0$, exista doar un K_{1D} pentru

care valorile proprii temporale sunt complex conjugate. Acest lucru se verifica si prin faptul ca $K_{1Dleft}=K_{1Dright}$.
Din relatia precedenta se vede ca se poate muta pozitia pe orizontala a punctului de mijloc al zonei cu valori proprii temporale complex conjugate.
Se vede ca atunci când $f_u=g_v$ punctul din mijlocul acestei zone este chiar originea.

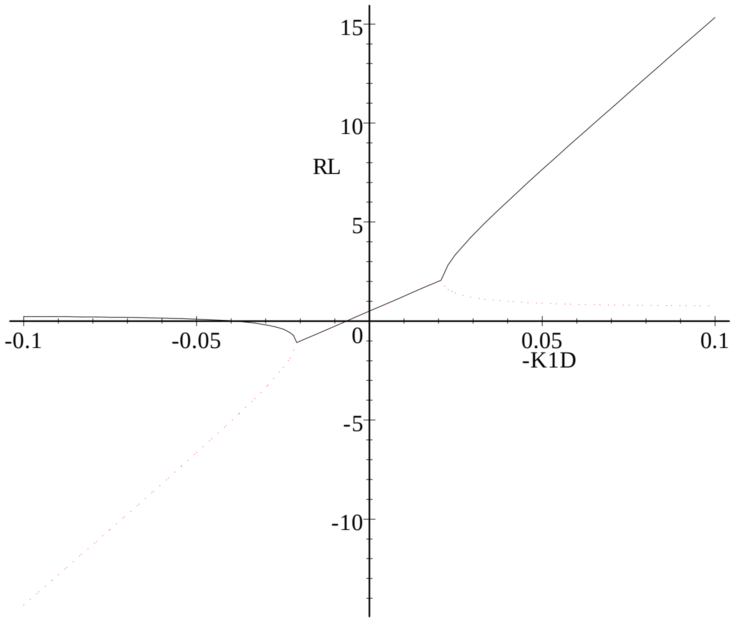


Fig. 43: Partea reala a valorilor proprii temporale functie de valoarea proprie spatiala (aceeasi pentru ambele straturi), cazul $f_u=g_v$

Partea reala a valorii proprii temporale pentru acest caz în origine este:

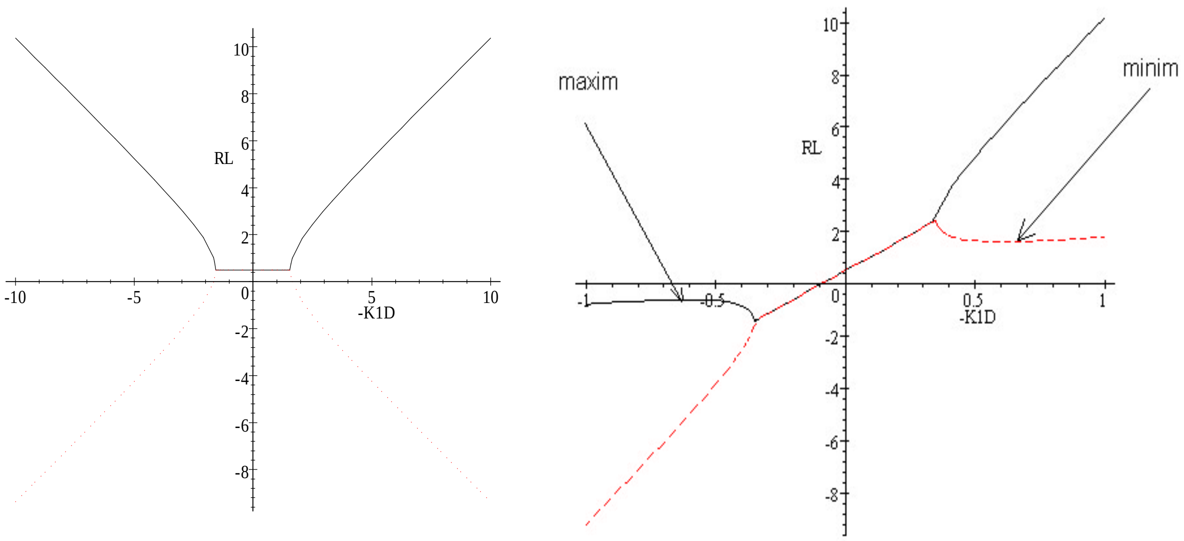
$$\mathrm{Re}(l_{1,2}(K_{1D}))=gf_u \tag{7.34.}$$

Atunci când D_u este diferit de $-D_v$ si exista doua extreme ($f_v g_u D_u D_v<0$), un maxim si un minim, unul pe locul geometric al unei radacini, iar celalalt pe locul geometric al celeilalte radacini.
Abscisele extremelor sunt:

$$\begin{aligned}
K_{1Dextrem1} &= \frac{g}{D_u-D_v}\left(g_v - f_u - (D_u+D_v)\sqrt{\frac{-f_v g_u}{D_u D_v}}\right)\\
K_{1Dextrem2} &= \frac{g}{D_u-D_v}\left(g_v - f_u + (D_u+D_v)\sqrt{\frac{-f_v g_u}{D_u D_v}}\right)
\end{aligned} \tag{7.35.}$$

Extremele sunt utile pentru obtinerea caracteristicilor de tip filtru trece banda si respectiv pentru obtinerea de caracteristici de tip filtru opreste banda.

Se reprezinta mai jos cazul în care a) Du=-Dv si b) cazul în care ele sunt diferite si în plus este îndeplinita si conditia $D_u D_v>0$ (cu $f_v g_u<0$ pentru a) si b)).



a) b)

Fig. 44: Partea reala a valorilor proprii temporale în functie de valoarea proprie spatiala (aceeasi pentru ambele straturi), ilustrarea notiunii de punct de extrem

Exemplu de proiectare

Se cere sa se proiecteze un filtru trece sus cu frecventa de taiere m_T data. Pentru acest caz se porneste de la situatia D_u=-D_v si $f_v g_u<0$. În plus, se alege un $f_u<0$ pentru a asigura o situatie ca în figura de mai jos:

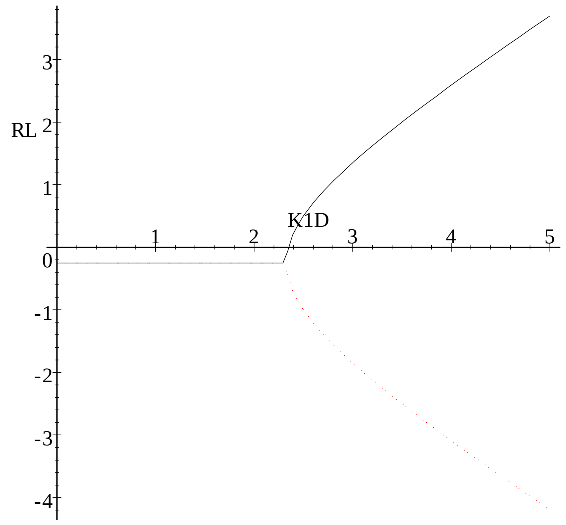


Fig. 45: Cazul D_u=-D_v, $f_u<0$, $f_v g_u<0$,

Se obtin valorile pentru: K_{1Dleft}=-0.83114 iar $K_{1Dright}$=2.3311.
Urmatorul pas este sa se izoleze din curba de dispersie zona care de interes astfel încât sa fie îndeplinita cerinta legata de m_T.

Se doreste proiectarea unei retele de M celule cu frontiere de tip zero flux. Atunci când se schimba conditiile de frontiera situatia nu se schimba semnificativ (a se consulta tabelul cu valori proprii spatiale pentru toate conditiile de granita). Este valabila urmatoarea ecuatie (se doreste ca parametrii template-ului A_0 si A_1 sa fie pozitivi):

$$\mathrm{Re}(K_T) = 0$$
$$K_T = A_0 - 2A_1 \cos\frac{p}{M} m_T \quad (7.36.)$$

Se va folosi tehnica ferestrei pentru a determina parametrii template-ului. Procedura este urmatoarea:

- se fixeaza largimea ferestrei avându-se grija sa nu se iasa spre stânga din zona cu valori proprii temporale complex conjugate. În cazul de fata o valoare A_1=1 este acceptabila;
- se cunoaste K_T=2.3508, m_T=7 si A_1=1 în acest moment. Se poate determina parametrul A_0 luând în considerare o retea de M=30 celule:

$$A_0 = K_T + 2A_1 \cos\frac{p}{M} m_T \quad (7.37.)$$

Valoarea determinata este A_0=3.8371
Curba de dispersie în functie de moduri este reprezentata în Fig. 46:

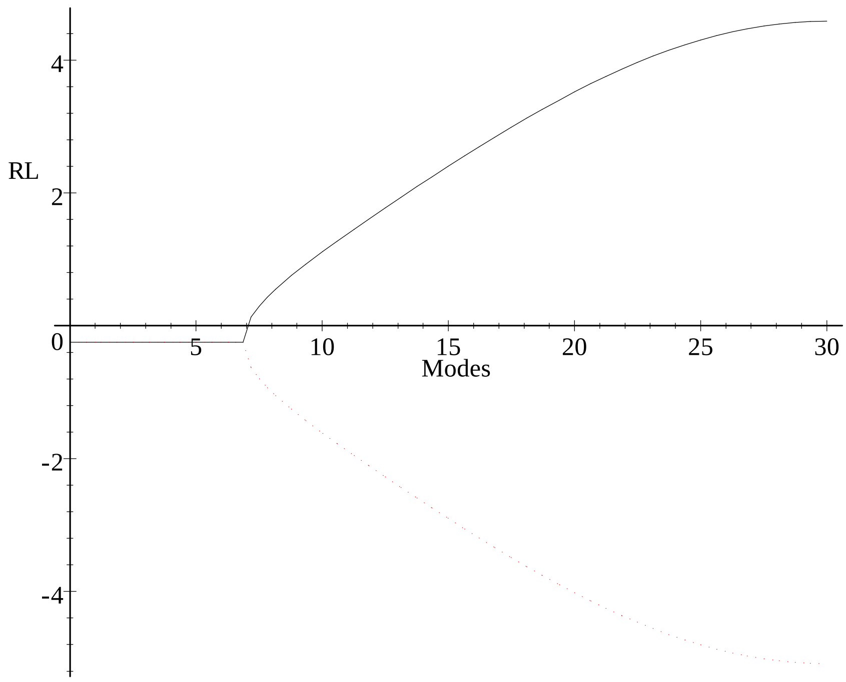


Fig. 46: FTS cu frecventa de taiere m=7, A_1=1, A_0=3.8371

Valorile parametrilor celulelor precum si parametrii retelei sunt dati mai jos: γ=5, f_u=-0.2, g_u=0.1, g_v=0.1, f_v=-1, D_u=1, D_v=-1 si respectiv A_0=3.84, A_1=1.

Observatii:

- daca semnul lui A_1 se schimba, va rezulta un filtru trece jos deoarece se schimba pozitia modurilor în raport cu K_{1D}. Asa-numita frecventa de taiere va fi la m=M-m_T=30-7=23. Astfel, daca se doreste proiectarea unui filtru trece jos se poate proiecta dupa metodologia expusa în acest exemplu un filtru trece sus având grija ca acesta sa se proiecteze având frecventa de taiere m_T=M-$m_{TFTJ.}$.
- aceste doua tipuri de filtre nu au nevoie de o curba de dispersie cu extreme. Daca se doreste proiectarea unui filtru trece banda sau a unuia opreste banda, atunci se va folosi clasa curbelor de dispersie care prezinta extreme, adica atunci când cele doua constante de difuzie pe cele doua straturi D_u si respectiv D_v nu sunt egale si de semn contrar.

În figura de mai jos se prezinta curba de dispersie a filtrului trece jos rezultat prin schimbarea semnului lui A_1.

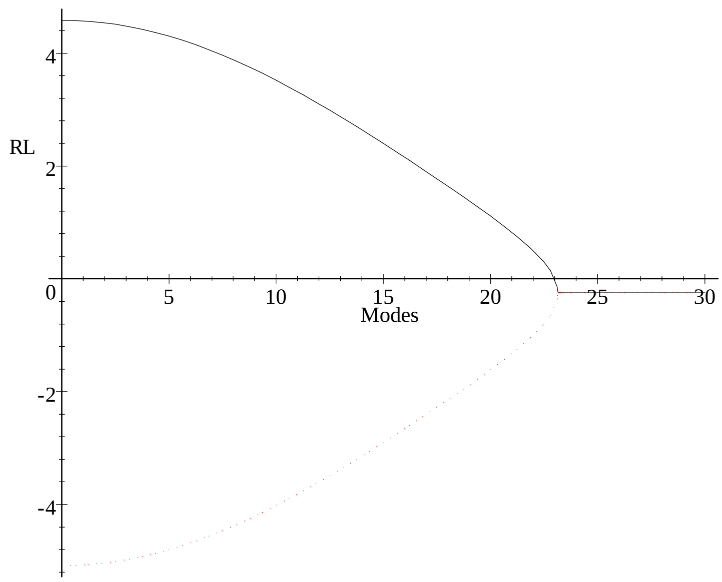


Fig. 47: FTJ cu “frecventa de taiere” m=23, A_1=-1, A_0=3.8371

Rezultate provenite din simulare

În cadrul acestei sectiuni se vor simula cele doua sisteme proiectate mai sus; în exemplul I va fi simulat filtrul trece sus, pe când în exemplul II se va simula comportarea filtrului trece jos.

Exemplul I

Pentru a verifica comportarea CNN-ului de tip filtrului trece sus, se va initializa în primul experiment legat de acest filtru spatio-temporal fiecare variabila de stare în asa fel încât acestea sa formeze modul 4 spatial, care este înafara "benzii de trecere" a filtrului respectiv.
Evolutia în timp a starii pe stratul notat cu u este:

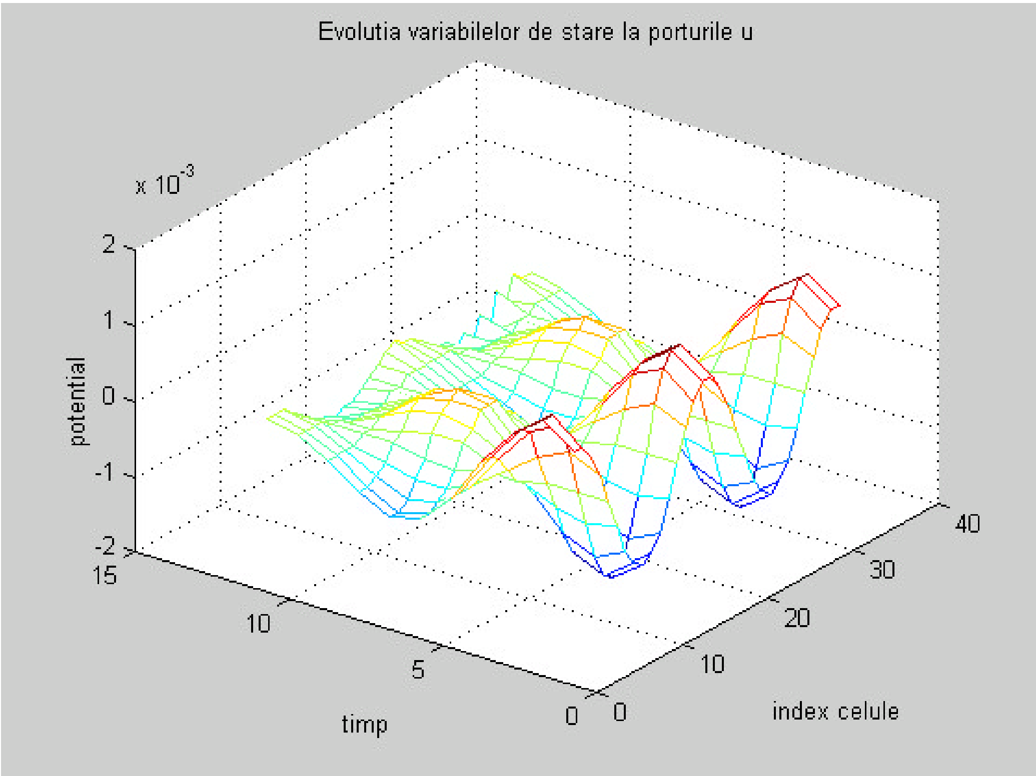


Fig. 48: Evolutia semnalului spatial

Se constata din evolutia semnalului spatial ca acesta tinde sa scada spre 0 pe masura ce trece timpul. Scaderea se face într-o maniera oscilanta, deoarece în zona modului 4, valorile proprii corespunzatoare modurilor sunt complex conjugate si cu parte reala negativa.
În continuare vom verifica faptul ca un mod care se afla în "banda de trecere" a filtrului trece sus este "amplificat" într-o maniera exponentiala.

În Fig. 49 se reprezinta evolutia modului 10 initializat cu valoarea 0.01.

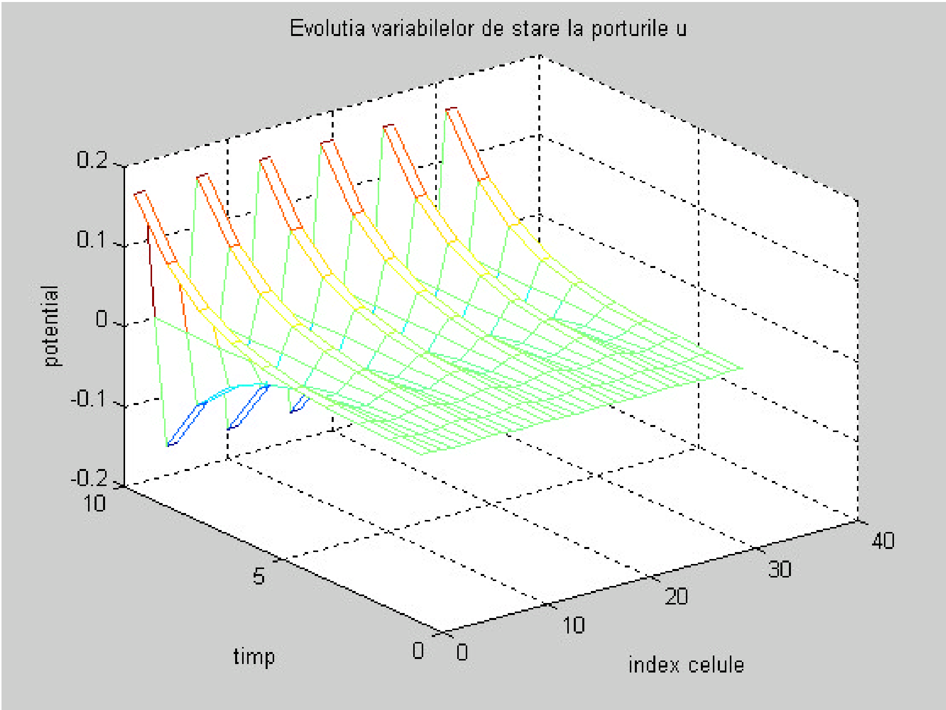
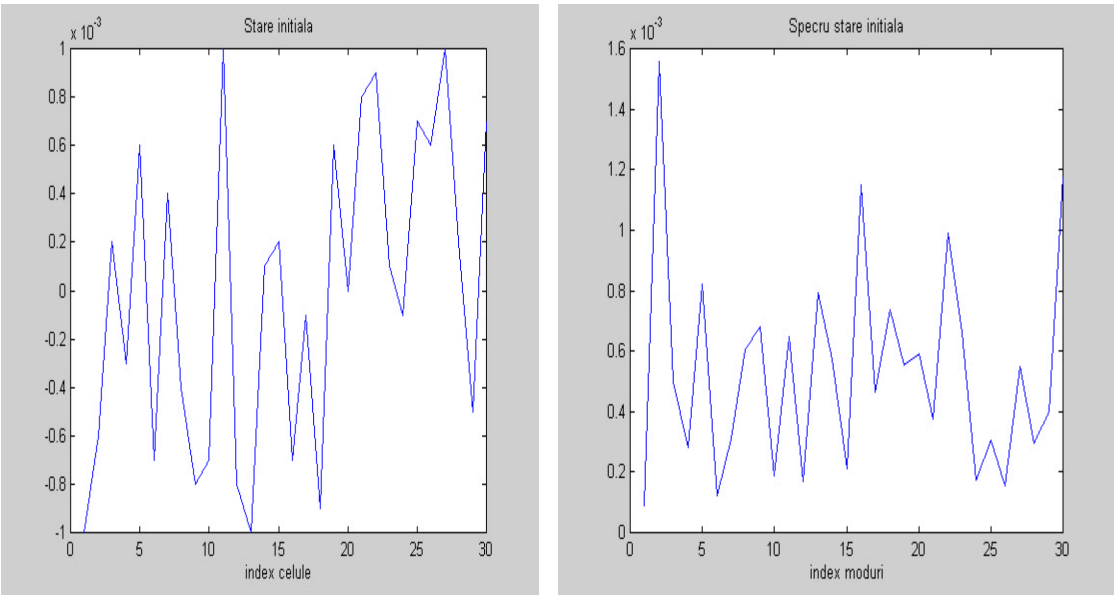


Fig. 49: Evolutia semnalului spatial pentru modul 10

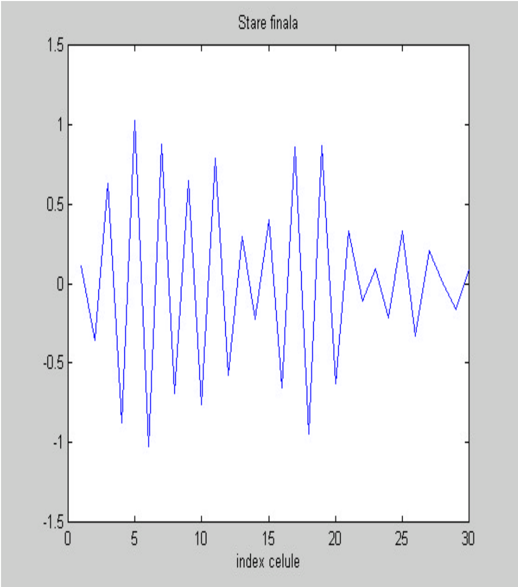
În cele din urma, se vor initializa variabilele de stare cu un semnal aleator pentru a realiza o filtrare trece sus a acestuia.
Filtrarea este destul de slaba (doar primele 7 moduri nu vor "trece" de acest filtru):



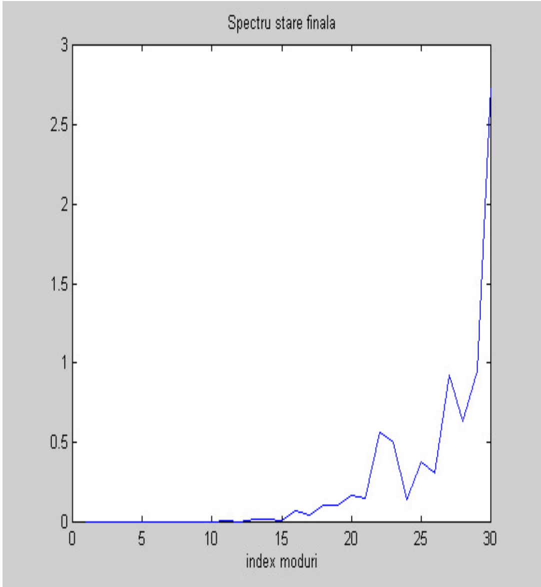
a) b)

Fig. 50: a) Stare initiala; b) Spectrul starii u initiale

Starea finala si spectrul acesteia se reprezinta mai jos:



a)



b)

Fig. 51: a) Stare finala; b) Spectru stare finala

observându-se caracterul trece-sus al filtrului.

Exemplul II

În acest exemplu se lucreaza cu filtrul spatial rezultat din schimbarea semnului lui A_1. Se initializeaza pentru început variabilele de stare de pe stratul u cu modul 25 de amplitudine 0.5.

Evolutia în timp a starii pe stratul u este reprezentata mai jos:

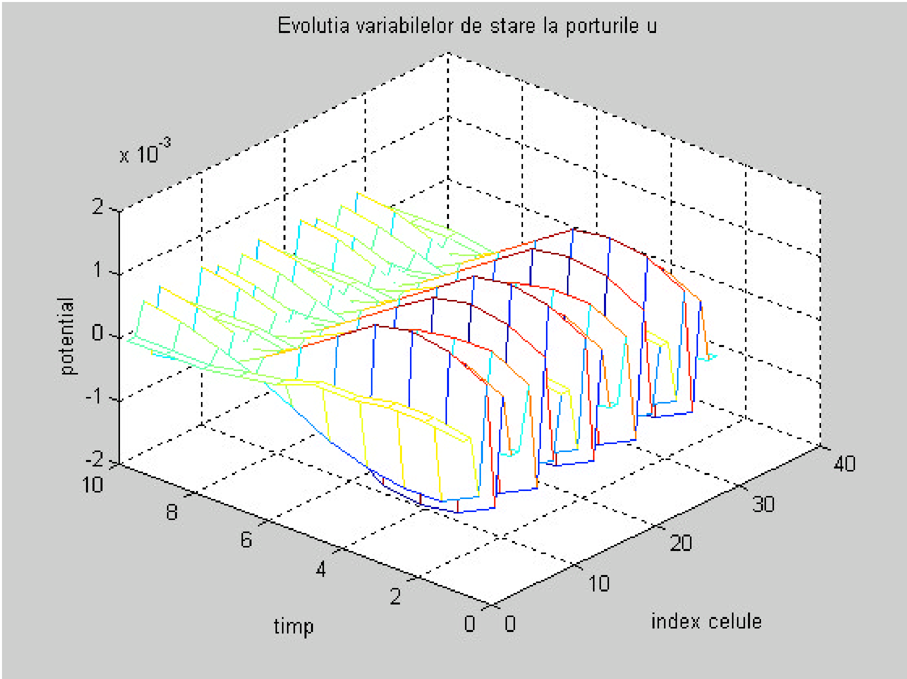


Fig. 52: Evolutia semnalului spatial în cazul initializarii cu modul 25

Daca se initializeaza reteaua cu modul 20 (în "banda de trecere"), se obtine evolutia:

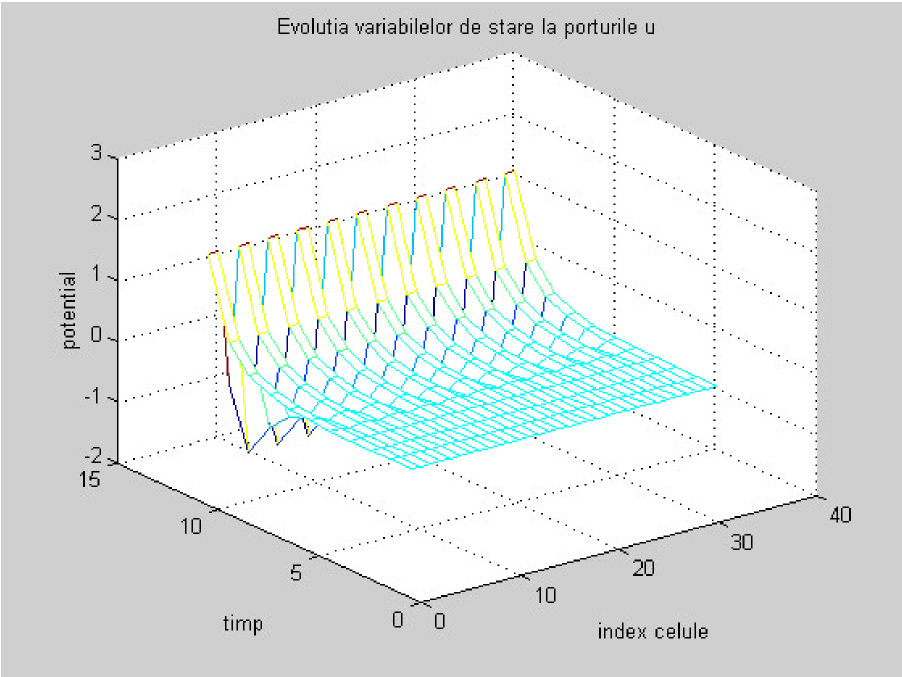
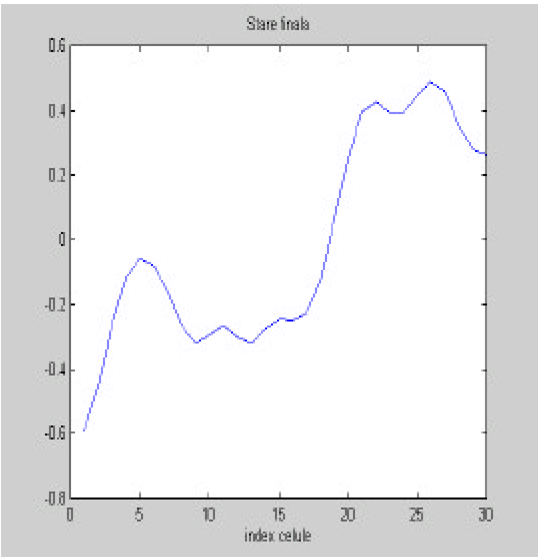
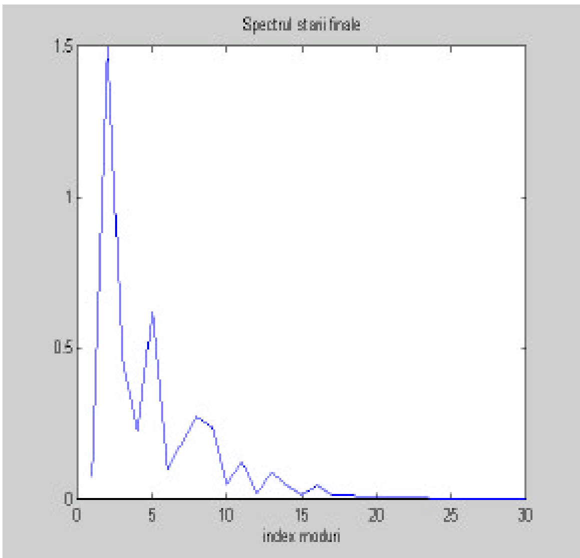


Fig. 53: Evolutia semnalului spatial în cazul initializarii cu modul 20

În sfârsit, se initializeaza reteaua cu un semnal aleator. Starea initiala si spectrul semnalului initial sunt aceleasi ca cele din Fig. 50:
Starea finala si respectiv spectrul starii finale sunt reprezentate mai jos:





a) b)

Fig. 54: a) Stare finala; b) Spectru stare finala

Capitolul 8.: Studiul retelei neuronale celulare 2D formate din celule de ordinul II conectate cu o vecinatate de raza unitate

Prin dubla interconectare a celulelor generale de ordinul II (solutia de circuit) se poate obtine un sistem autonom a carui functionare în zona liniara centrala a celulelor liniare pe portiuni sa poata realiza o procesare de imagine utila.

Interconexiunile

Modul de interconectare este similar cazului 1D prezentat în capitolul 7, interconexiunile fiind aici descrise de un template bidimensional care, în general este nesimetric.
Coeficientii acestuia au fost notati cu $A_{i,j}$ cu fiecare dintre indici luând valori în multimea {-1, 0, 1}.

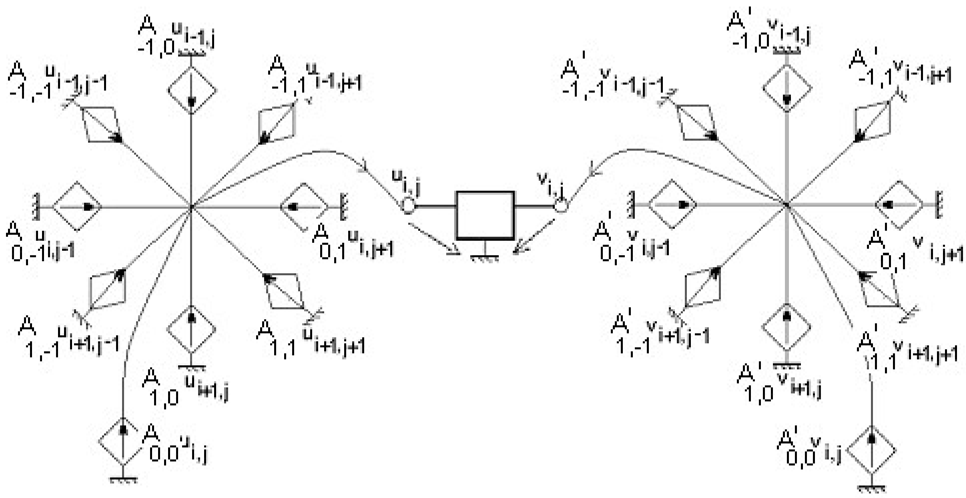


Fig 55: Interconexiunile în CNN-ul 2D

S-a considerat, ca si în cazul 1D ca cele doua straturi au template-uri diferite.

Template-uri asimetrice si template-uri simetrice

În cazul cel mai general se vorbeste de template-uri asimetrice. Expresia operatorului care descrie template-ul este:

$$\begin{aligned} O_{2D}(u_{i,j}) = {} & A_{-1,-1}u_{i-1,j-1} + A_{-1,0}u_{i-1,j} + A_{-1,1}u_{i-1,j+1} + A_{0,-1}u_{i,j-1} + \\ & + A_{0,0}u_{i,j} + A_{0,1}u_{i,j+1} + A_{1,-1}u_{i+1,j-1} + A_{1,0}u_{i+1,j} + A_{1,1}u_{i+1,j+1} \end{aligned} \qquad (8.1.)$$

Pentru cel de-al doilea strat apare semnul „prim”.
Dupa cum s-a specificat si mai sus, template-urile corespunzatoare celor doua straturi sunt în general diferite. Se numeste simetric un template pentru care parametrii $A_{-1,-1}$, $A_{1,1}$, $A_{1,-1}$ si $A_{-1,1}$, $A_{0,-1}$ si $A_{0,1}$, $A_{-1,0}$ si $A_{1,0}$ sunt egali.

Conditii de granita

Consideratiile prezentate în capitolul 7 referitoare la conditiile de granita se extind imediat la cazul 2D, pentru fiecare directie putând exista conditii de granita precum cele discutate în cazul 1D. Se vor trece succint în revista câteva cazuri, remarcând ca sunt posibile conditii la limita diferite pentru fiecare latura a CNN-ului (desigur cu exceptia cazului periodic).

Pentru un CNN cu MxN celule, conditiile de granita sunt descrise prin intermediul urmatoarelor seturi de ecuatii:

- *conditii de granita de tip periodic (inel)*

$$\begin{aligned} &u(-1, j,t) = u(M-1, j,t),\ u(i,-1,t) = u(i, N-1,t),\\ &u(0, j,t) = u(M, j,t),\ u(i,0,t) = u(i, N,t) \end{aligned} \quad (8.2.)$$

- *conditii de granita de tip zero-flux*

$$\begin{aligned} &u(-1, j,t) = u(0, j,t),\ u(i,-1,t) = u(i,0,t),\\ &u(M, j,t) = u(M-1, j,t),\ u(i, N,t) = u(i, N-1,t) \end{aligned} \quad (8.3.)$$

- *conditii de granita de tip anti-zero-flux*

$$\begin{aligned} &u(-1,j,t) = -u(0,j,t),\ u(M,j,t) = -u(M-1,j,t)\\ &u(i,-1,t) = -u(i,0,t),\ u(i,N,t) = -u(i,N-1,t) \end{aligned} \quad (8.4.)$$

- *conditii de granita de tip zero*

$$\begin{aligned} &u(-1,j,t) = 0,\ u(M,j,t) = 0\\ &u(i,-1,t) = 0,\ u(i,N,t) = 0 \end{aligned} \quad (8.5.)$$

- *conditii de granita de tip quasi-zero flux*

$$\begin{aligned} &u(-1,j,t) = u(1,j,t),\ u(M,j,t) = u(M-2,j,t)\\ &u(i,-1,t) = u(i,1,t),\ u(i,N,t) = u(i,N-2,t) \end{aligned} \quad (8.6.)$$

Pentru toate conditiile de granita aceleasi relatii sunt valabile si pentru porturile etichetate cu v.

Pe lânga conditiile de granita aceleasi pentru toate granitele se mai pot mentiona conditii de granita mixte, cum ar fi:

- *conditii de granita quasizero-flux-quasizero-flux zero-zero*

$$\begin{aligned} &u(-1,j,t) = 0,\ u(M,j,t) = 0\\ &u(i,-1,t) = u(i,1,t),\ u(i,N,t) = u(i,N-2,t)\\ &v(-1,j,t) = 0,\ v(M,j,t) = 0\\ &v(i,-1,t) = v(i,1,t),\ v(i,N,t) = v(i,N-2,t) \end{aligned} \quad (8.7.)$$

- *conditii de granita zero-zero-flux: antizero-flux-quasi-zero-flux*

$$\begin{aligned} & u(i,-1,t) = 0,\ \ v(i,-1,t) = 0 \\ & u(i,N,t) = u(i,N-1,t),\ \ v(i,N,t) = v(i,N-1,t) \\ & u(-1,j,t) = -u(0,j,t),\ \ v(-1,j,t) = -v(0,j,t) \\ & u(M,j,t) = u(M-2,j,t),\ \ v(M,j,t) = v(M-2,j,t) \end{aligned} \tag{8.8.}$$

Ecuatii si solutie

Forma generala a ecuatiilor care descriu functionarea sistemului format din celule de ordinul 2, dublu cuplate este dat în relatiile de mai jos:

$$\begin{cases} \dfrac{du_{i,j}(t)}{dt} = gf(u_{i,j}, v_{i,j}) + D_u O_{2D}(u_{i,j}) & i = 0..M-1 \\ \dfrac{dv_{i,j}(t)}{dt} = gg(u_{i,j}, v_{i,j}) + D_v O_{2D}(v_{i,j}) & j = 0..N-1 \end{cases} \tag{8.9.}$$

În cele ce urmeaza se prezinta succint modul de solutionare al sistemului de ecuatii (8.9) ca o generalizare a metodei din capitolul 7. Astfel, în ipoteza liniarizarii sistemul de ecuatii devine

$$\begin{cases} \dfrac{du_{i,j}(t)}{dt} = g\,(f_u u_{i,j} + f_v v_{i,j}) + D_u O_{2D}(u_{i,j}) & i = 0..M-1 \\ \dfrac{dv_{i,j}(t)}{dt} = g\,(g_u u_{i,j} + g_v v_{i,j}) + D_v O_{2D}(v_{i,j}) & j = 0..N-1 \end{cases} \tag{8.10.}$$

unde notatiile au aceeasi semnificatie ca în capitolul 7

Utilizând schimbarea de variabila

$$\begin{cases} u_{i,j} = \displaystyle\sum_{m=0}^{M-1}\sum_{n=0}^{N-1} \Phi_{MN}(m,n,i,j)\,\hat{u}_{mn} & i = 0..M-1 \\ v_{i,j} = \displaystyle\sum_{m=0}^{M-1}\sum_{n=0}^{N-1} \Phi_{MN}(m,n,i,j)\,\hat{v}_{mn} & j = 0..N-1 \end{cases} \tag{8.11.}$$

careia îi corespund formulele de inversiune

$$\begin{cases} \hat{u}_{mn} = \displaystyle\sum_{m=0}^{M-1}\sum_{n=0}^{N-1} \Phi^{*}{}_{MN}(m,n,i,j)\,u_{i,j} & i = 0..M-1 \\ \hat{v}_{mn} = \displaystyle\sum_{m=0}^{M-1}\sum_{n=0}^{N-1} \Phi^{*}{}_{MN}(m,n,i,j)\,v_{i,j} & j = 0..N-1 \end{cases} \tag{8.12.}$$

Cel mai general caz cu care vom lucra este cazul în care functiile Φ sunt produse de exponentiale.

$$\Phi_{MN}(w_x(m), j_x(m), w_y(n), j_y(n), i, j) = e^{j(w_x(m)i + j_x(m))} e^{j(w_y(n)j + j_y(n))} \tag{8.13.}$$

Functia din relatia (8.13) este functie proprie pentru operatorul O_{2D}, iar $K_{2D}(m,n)$ este valoare proprie în raport cu operatorul mentionat.

$$\begin{aligned} O_{2D}(\Phi_M(w_x(m), j_x(m), w_y(n), j_y(n), i, j)) &= \\ K_{2D}(m,n)\Phi_M(w_x(m), j_x(m), w_y(n), j_y(n), i, j) \end{aligned} \tag{8.14.}$$

Demonstratie
Demonstratia se face în mod similar cu cea pentru cazul 1D, scotând în factor comun produsul de exponentiale din relatia (8.13).
Dupa gruparea numarului complex ce iese în factor comun în parte reala si parte imaginara, rezulta valoarea proprie:

$$\begin{aligned} K_{2D}(m,n) = A_{0,0} &+ \\ &\left(A_{-1,-1} + A_{1,1}\right)\cos\left(w_x(m) + w_y(n)\right) + \left(A_{1,-1} + A_{-1,1}\right)\cos\left(w_x(m) - w_y(n)\right) + \\ &\left(A_{-1,0} + A_{1,0}\right)\cos\left(w_x(m)\right) + \left(A_{0,-1} + A_{0,1}\right)\cos\left(w_y(n)\right) + \\ &j\Big\{\left(A_{1,1} - A_{-1,-1}\right)\sin\left(w_x(m) + w_y(n)\right) + \left(A_{1,-1} - A_{-1,1}\right)\sin\left(w_x(m) - w_y(n)\right) + \\ &\left(A_{1,0} - A_{-1,0}\right)\sin\left(w_x(m)\right) + \left(A_{0,1} - A_{0,-1}\right)\sin\left(w_y(n)\right)\Big\} \end{aligned} \tag{8.15.}$$

Parcurgând aceleasi etape ca în cazul 1D, se deduc pe rând: setul de ecuatii decuplate (tinând seama ca în cazul 2D expresia lui $K_{1D}(m)$-> $K_{2D}(m,n)$ iar indicele i->(i,j)) se rezolva în acelasi mod si se obtine solutia sistemului de ecuatii:

$$u_{i,j}(t) = \sum_{m=0}^{M-1}\sum_{n=0}^{N-1}\left\{\begin{aligned} &a_{mn}(\hat{e}_{mn}, \hat{e}_{mn}', \hat{u}_{mn}(0), \hat{v}_{mn}(0))e^{\lambda_{mn1}t} + \\ &b_{mn}(\hat{e}_{mn}, \hat{e}_{mn}', \hat{u}_{mn}(0), \hat{v}_{mn}(0))e^{\lambda_{mn2}t} + f_1(\hat{e}_{mn}, \hat{e}_{mn}') \end{aligned}\right\}\Phi_{MN}(m,n,i,j) \tag{8.16.}$$

$$v_{i,j}(t) = \sum_{m=0}^{M-1}\sum_{n=0}^{N-1}\left\{\begin{aligned} &c_{mn}(\hat{e}_{mn}, \hat{e}_{mn}', \hat{u}_{mn}(0), \hat{v}_{mn}(0))e^{\lambda_{mn1}t} + \\ &d_{mn}(\hat{e}_{mn}, \hat{e}_{mn}', \hat{u}_{mn}(0), \hat{v}_{mn}(0))e^{\lambda_{mn2}t} + f_2(\hat{e}_{mn}, \hat{e}_{mn}') \end{aligned}\right\}\Phi_{MN}(m,n,i,j)$$

Parametrii λ_{mn1} si λ_{mn2} valorile proprii temporale ce caracterizeaza modul spatial (m,n).

cu

$$a_{mn}(\hat{e}_{mn},\hat{e}_{mn}^{'},\hat{u}_{mn}(0),\hat{v}_{mn}(0))=$$
$$-\frac{1}{p_{mn}q_{mn}-1}\hat{u}_{mn}(0)+\frac{q_{mn}}{p_{mn}q_{mn}-1}\hat{v}_{mn}(0)+\frac{f_1(\hat{e}_{mn},\hat{e}_{mn}^{'})-q_{mn}f_2(\hat{e}_{mn},\hat{e}_{mn}^{'})}{p_{mn}q_{mn}-1}$$

$$d_{mn}(\hat{e}_{mn},\hat{e}_{mn}^{'},\hat{u}_{mn}(0),\hat{v}_{mn}(0))=$$
$$\frac{p_{mn}}{p_{mn}q_{mn}-1}\hat{u}_{mn}(0)-\frac{1}{p_{mn}q_{mn}-1}\hat{v}_{mn}(0)+\frac{f_2(\hat{e}_{mn},\hat{e}_{mn}^{'})-p_{mn}f_1(\hat{e}_{mn},\hat{e}_{mn}^{'})}{p_{mn}q_{mn}-1}$$

$$b_{mn}(\hat{e}_{mn},\hat{e}_{mn}^{'},\hat{u}_{mn}(0),\hat{v}_{mn}(0))=$$
$$\frac{p_{mn}q_{mn}}{p_{mn}q_{mn}-1}\hat{u}_{mn}(0)-\frac{q_{mn}}{p_{mn}q_{mn}-1}\hat{v}_{mn}(0)+\frac{f_2(\hat{e}_{mn},\hat{e}_{mn}^{'})-p_{mn}f_1(\hat{e}_{mn},\hat{e}_{mn}^{'})}{p_{mn}q_{mn}-1}q_{mn}$$

$$c_{mn}(\hat{e}_{mn},\hat{e}_{mn}^{'},\hat{u}_{mn}(0),\hat{v}_{mn}(0))=$$
$$-\frac{p_{mn}}{p_{mn}q_{mn}-1}\hat{u}_{mn}(0)+\frac{p_{mn}q_{mn}}{p_{mn}q_{mn}-1}\hat{v}_{mn}(0)+\frac{f_1(\hat{e}_{mn},\hat{e}_{mn}^{'})-q_{mn}f_2(\hat{e}_{mn},\hat{e}_{mn}^{'})}{p_{mn}q_{mn}-1}p_{mn}$$

$$p_{mn}=\frac{l_{mn1}-gf_u-D_uK_{2D}(m,n)}{gf_v};\; q_{mn}=\frac{l_{mn2}-gf_u-D_vK_{2D}^{'}(m,n)}{gg_u}$$

si

$$\begin{cases} f_1(\hat{e}_{mn},\hat{e}_{mn}^{'})=\dfrac{-g\,(gg_v+K_{2D}^{'}D_v)\hat{e}_{mn}+g^2f_v\hat{e}_{mn}^{'}}{(gf_u+K_{2D}D_u)(gg_v+K_{2D}^{'}D_v)-g^2f_vg_u} \\ f_2(\hat{e}_{mn},\hat{e}_{mn}^{'})=\dfrac{g^2g_u\hat{e}_{mn}-g\,(gf_u+K_{2D}D_u)\hat{e}_{mn}^{'}}{(gf_u+K_{2D}D_u)(gg_v+K_{2D}^{'}D_v)-g^2f_vg_u} \end{cases}$$

Din forma generala a solutiei, se poate vedea ca depinde de conditiile initiale, valorile curentilor debitati de sursele independente de curent, valorile proprii temporale si de forma functiilor proprii, care este legata de conditiile de granita, dupa cum se va vedea în continuare.
Folosind o combinatie potrivita de conditii initiale pe straturile u si v se poate realiza suprimarea unor moduri spatiale [72]. Este de interes suprimarea unora dintre acelea cu valori proprii temporale cu parte reala pozitiva.

Functii proprii si valori proprii pentru exponentiale (conditii de frontiera periodice si template-uri asimetrice)

Pentru cazul general valorile proprii spatiale sunt numere complexe. În cazul template-ului simetric dupa definitia din acest capitol (nu în mod *necesar* cu simetrie *radiala*!) valoarea proprie spatiala este un numar real.
Valoarea proprie este în acest caz data de relatia:

$$K_{2D}=A_{0,0}+2A_{1,1}\cos\big(w_x(m)+w_y(n)\big)+2A_{-1,1}\cos\big(w_x(m)-w_y(n)\big)+ \quad (8.17.)$$
$$2A_{-1,0}\cos\big(w_x(m)\big)+2A_{0,-1}\cos\big(w_y(n)\big)$$

Demonstratie:
Afirmatia se demonstreaza usor analizând cazurile pentru care partea imaginara a expresiei valorii proprii spatiale din relatia (8.15) se anuleaza.

Suprafata de dispersie pentru cazul exponentialelor

Conditia necesara si suficienta pentru ca sistemul sa devina instabil este, similar cazului 1D, ca acesta sa aiba cel putin o componenta spectrala în semnalul constituit de conditiile initiale cu partea reala a valorii proprii temporale corespunzatoare pozitiva:

$$\mathrm{Re}(l_{1,2}(K_{2D},K_{2D}^{'})) = \mathrm{Re}\{g\,\frac{f_u+g_v}{2}+\frac{D_uK_{2D}+D_vK_{2D}^{'}}{2} \pm\sqrt{\left[g\,\frac{(g_v-f_u)}{2}-\frac{(D_uK_{2D}-D_vK_{2D}^{'})}{2}\right]^2+g^2 f_v g_u}\,\} \tag{8.18.}$$

Toate celelalte concepte referitoare la aceasta curba pentru cazul 1D se regasesc pentru cazul 2D, o trecere în revista a proprietatilor nemaifiind necesara la acest moment.

Decuplarea straturilor

Se observa ca în cazul în care termenul $\gamma^2 f_v g_u$ este mic (sau zero) acesta se poate neglija si se obtine pentru acest caz special un sistem format din doua CNN-uri cu celule de ordinul 1, care functioneaza independent unul fata de celalalt. Pentru aceasta situatie particulara, valorile proprii devin:

$$\begin{aligned} l_{mn1} &\cong gf_u + D_uK_{2D}(m,n) \\ l_{mn2} &\cong gg_v + D_vK_{2D}^{'}(m,n) \end{aligned} \tag{8.19.}$$

Prima valoare proprie depinde numai de parametrii asociati primului strat (asociat cu indexul u), în timp ce a doua radacina depinde numai de cei corespunzatori celui de-al doilea (asociat cu indexul v). Curba de dispersie corespunzând fiecarui sub-sistem separat format din celule de ordinul I este o dreapta cu panta depinzând de semnul lui D_u si respectiv D_v. Aceasta corespunde rezultatelor care s-au raportat pâna în acest moment pentru CNN-uri cu celule de ordinul I.

Template-uri simetrice

În ceea ce urmeaza ne vom îndrepta atentia spre template-urile simetrice. Dupa cum am mai mentionat, pentru aceasta familie de template-uri (cele simetrice) decuplarea ecuatiilor este posibila si pentru alte tipuri de conditii de granita si în consecinta functii proprii. Functiile proprii sunt în toate aceste cazuri reale si tot reale vor fi si valorile proprii spatiale.

Se considera template-ul:

$A_{1,1}$	$A_{-1,0}$	$A_{-1,1}$
$A_{0,-1}$	$A_{0,0}$	$A_{0,-1}$
$A_{-1,1}$	$A_{-1,0}$	$A_{1,1}$

Valoarea proprie asociata functiei proprii de tip produs de exponentiale este aceeasi si pentru functiile proprii date în relatia (8.20) daca si $A_{1,1}$=$A_{-1,1}$.

$$\begin{aligned}\Phi_{MN}(-w_x(m),-j_x(m),-w_y(n),-j_y(n),i,j)&=e^{-j(w_x(m)i+j_x(m))}e^{-j(w_y(n)j+j_y(n))}\\ \Phi_{MN}(-w_x(m),-j_x(m),w_y(n),j_y(n),i,j)&=e^{-j(w_x(m)i+j_x(m))}e^{j(w_y(n)j+j_y(n))}\end{aligned} \tag{8.20.}$$

Demonstratie:
Valoarea proprie corespunzatoare functiei proprii produs de exponentiale este data în relatia (8.15):
În ipoteza în care este $A_{1,1}$ este diferit de $A_{-1,1}$, se vede ca functiile proprii luate în discutie au valori proprii (în raport cu operatorul liniar spatial) diferite. Presupunem ca $A_{1,1}$= $A_{-1,1}$.
Valoarea proprie devine:

$$\begin{aligned}K_{2D} = A_{0,0} + 2A_{1,1}\left(\cos\left(w_x(m)+w_y(n)\right)+\cos\left(w_x(m)-w_y(n)\right)\right)+\\ 2A_{-1,0}\cos\left(w_x(m)\right)+2A_{0,-1}\cos\left(w_y(n)\right)=\\ A_{0,0}+4A_{1,1}\cos\left(w_x(m)\right)\cos\left(w_y(n)\right)+\\ 2A_{-1,0}\cos\left(w_x(m)\right)+2A_{0,-1}\cos\left(w_y(n)\right)\end{aligned} \tag{8.21.}$$

Se vede ca valoarea proprie prezinta simetrie para atât în raport cu $\omega_x(m)$ cât si cu $\omega_y(n)$. Prin urmare, afirmatia este demonstrata.

Functiile:

$$\begin{aligned}\Phi_{MN}'\left(w_x(m),j_x(m),w_y(n),j_y(n),i,j\right)&=\\ \cos\left(w_x(m)i+j_x(m)\right)&\cos\left(w_y(n)j+j_y(n)\right)\\ \Phi_{MN}''\left(w_x(m),j_x(m),w_y(n),j_y(n),i,j\right)&=\\ \sin\left(w_x(m)i+j_x(m)\right)&\sin\left(w_y(n)j+j_y(n)\right)\\ \Phi_{MN}'''\left(w_x(m),j_x(m),w_y(n),j_y(n),i,j\right)&=\\ \sin\left(w_x(m)i+j_x(m)\right)&\cos\left(w_y(n)j+j_y(n)\right)\end{aligned} \tag{8.22.}$$

sunt functii proprii pentru operatorul O_{2D} si corespunzator au aceeasi valoare proprie spatiala ca si functia exponentiala pentru cazul template-ului simetric în care $A_{1,1}= A_{-1,1}$:

Demonstratie:
Se demonstreaza prima relatie:

$$
\begin{aligned}
&O_{2D}\left(\Phi_{MN}{}'\left(w_x(m), j_x(m), w_y(n), j_y(n), i, j\right)\right) = \\
&\quad \frac{1}{4}\Big\{O_{2D}\left(\Phi_{MN}\left(w_x(m), j_x(m), w_y(n), j_y(n), i, j\right)\right) + \\
&\quad\quad O_{2D}\left(\Phi_{MN}\left(-w_x(m), -j_x(m), w_y(n), j_y(n), i, j\right)\right) + \\
&\quad\quad O_{2D}\left(\Phi_{MN}\left(w_x(m), j_x(m), -w_y(n), -j_y(n), i, j\right)\right) + \\
&\quad\quad O_{2D}\left(\Phi_{MN}\left(-w_x(m), -j_x(m), -w_y(n), -j_y(n), i, j\right)\right)\Big\} = \\
&\quad \frac{1}{4}\Big\{K_{2D}\Phi_{MN}\left(w_x(m), j_x(m), w_y(n), j_y(n), i, j\right) + \\
&\quad\quad K_{2D}\Phi_{MN}\left(-w_x(m), -j_x(m), w_y(n), j_y(n), i, j\right) + \\
&\quad\quad K_{2D}\Phi_{MN}\left(w_x(m), j_x(m), -w_y(n), -j_y(n), i, j\right) + \\
&\quad\quad K_{2D}\Phi_{MN}\left(-w_x(m), -j_x(m), -w_y(n), -j_y(n), i, j\right)\Big\} = \\
&\quad\quad\quad\quad K_{2D}\Phi_{MN}{}'\left(w_x(m), j_x(m), w_y(n), j_y(n), i, j\right)
\end{aligned}
$$

A doua si a treia relatie se pot demonstra imediat daca se face observatia:

$$
\begin{aligned}
j_x(m) &\to j_x(m) - \frac{p}{2} \\
j_y(n) &\to j_y(n) - \frac{p}{2}
\end{aligned}
\tag{8.23.}
$$

Conditii de granita speciale

Conditiile de granita despre care s-a discutat în sectiunea *Conditii de granita* corespund fiecare unui anumit tip de functie proprie [69]. Conditiile de granita pot fi *pure* si respectiv *impure*. Se vorbeste de conditii de granita *pure* când se lucreaza cu acelasi tip de conditii de granita la fiecare latura a retelei celulare 2D.
Când se vorbeste de conditii de granita diferite pentru fiecare latura a retelei, se va spune ca aceasta situatie corespunde conditiilor de granita *impure*.
Tabelul ce urmeaza este similar cu cel de la cazul 1D, specificul fiind numarul mai mare de posibilitati.

Tipul conditiei de granita	$\Phi_{MN}(\omega_x(m),\varphi_x(m), \omega_y(n), \varphi_y(n),i,j)$	$\omega_x(m)$	$\varphi_x(m)$	$\omega_y(n)$	$\varphi_y(n)$
azf-azf-azf-azf	$\sin\frac{(2i+1)(m+1)p}{2M}\sin\frac{(2j+1)(n+1)p}{2N}$	$\frac{p}{M}m$	$\frac{(m+1)p}{2M}\left\{-\frac{p}{2}\right\}$	$\frac{(n+1)p}{N}$	$\frac{(n+1)p}{2N}\left\{-\frac{p}{2}\right\}$
z-z-z-z	$\sin\frac{(i+1)(m+1)p}{M+1}\sin\frac{(j+1)(n+1)p}{2N}$	$\frac{p}{M+1}(m+1)$	$\frac{(m+1)p}{M+1}\left\{-\frac{p}{2}\right\}$	$\frac{(n+1)p}{N+1}$	$\frac{(n+1)p}{N+1}\left\{-\frac{p}{2}\right\}$
qzf-qzf-qzf-qzf	$\cos\frac{mip}{M-1}\cos\frac{njp}{N-1}$	$\frac{p}{M-1}m$	0	$\frac{np}{N-1}$	0
zf-zf-zf-zf	$\cos\frac{(2i+1)mp}{2M}\cos\frac{(2j+1)np}{2N}$	$\frac{p}{M}m$	$\frac{mp}{2M}$	$\frac{np}{N}$	$\frac{np}{2N}$
r-r-r-r	$e^{j\frac{2p}{M}mi}e^{j\frac{2p}{N}nj}$	$\frac{2p}{M}m$	0	$\frac{2np}{N}$	0
qzf-qzf-z-z	$\cos\frac{mip}{M-1}\sin\frac{(n+1)(j+1)p}{N+1}$	$\frac{p}{M-1}m$	0	$\frac{(n+1)p}{N+1}$	$\frac{(n+1)p}{N+1}\left\{-\frac{p}{2}\right\}$

azf-azf-z-z	$\sin\frac{(2i+1)(m+1)p}{2M}\sin\frac{(j+1)(n+1)p}{N+1}$	$\frac{p}{M}(m+1)$	$\frac{(m+1)p}{2M}\left\{-\frac{p}{2}\right\}$	$\frac{(n+1)p}{N+1}$	$\frac{(n+1)p}{N+1}\left\{-\frac{p}{2}\right\}$
z-z-zf-zf	$\sin\frac{(i+1)(m+1)p}{M+1}\cos\frac{(2j+1)np}{2N}$	$\frac{p}{M+1}(m+1)$	$\frac{(m+1)p}{M+1}\left\{-\frac{p}{2}\right\}$	$\frac{np}{N}$	$\frac{np}{2N}$
zf-zf-azf-azf	$\cos\frac{(2i+1)mp}{2M}\sin\frac{(2j+1)(n+1)p}{2N}$	$\frac{p}{M}m$	$\frac{mp}{2M}$	$\frac{(n+1)p}{N}$	$\frac{(n+1)p}{2N}\left\{-\frac{p}{2}\right\}$
zf-zf-qzf-qzf	$\cos\frac{(2i+1)mp}{2M}\cos\frac{njp}{N-1}$	$\frac{p}{M}m$	$\frac{mp}{2M}$	$\frac{np}{N-1}$	0
qzf-qzf-azf-azf	$\cos\frac{mip}{M-1}\sin\frac{(2j+1)(n+1)p}{2N}$	$\frac{p}{M-1}m$	0	$\frac{(n+1)p}{N}$	$\frac{(n+1)p}{2N}\left\{-\frac{p}{2}\right\}$
z-zf-azf-qzf	$\sin\frac{(i+1)(2m+1)p}{2M+1}\sin\frac{(2j+1)(2n+1)p}{2(2N-1)}$	$\frac{p}{2M+1}(2m+1)$	$\frac{(2m+1)p}{2M+1}\left\{-\frac{p}{2}\right\}$	$\frac{(2n+1)p}{2N-1}$	$\frac{(2n+1)p}{2(2N-1)}\left\{-\frac{p}{2}\right\}$

Tabelul 2: Functii proprii pentru diferite tipuri de conditii de granita (2D)

Tipul conditiei de granita	K_{2D}
azf-azf-azf-azf	$A_{0,0}+2A_{-1,0}\cos(\frac{p}{M}m)+4A_{1,1}\cos(\frac{p}{M}m)\cos\frac{(n+1)p}{N}+2A_{0,-1}\cos\frac{(n+1)p}{N}$
z-z-z-z	$A_{0,0}+2A_{-1,0}\cos(\frac{p}{M+1}(m+1))+4A_{1,1}\cos(\frac{p}{M+1}(m+1))\cos(\frac{(n+1)p}{N+1})+2A_{0,-1}\cos(\frac{(n+1)p}{N+1})$
qzf-qzf-qzf-qzf	$A_{0,0}+2A_{-1,0}\cos(\frac{p}{M-1}m)+4A_{1,1}\cos(\frac{p}{M-1}m)\cos(\frac{np}{N-1})+2A_{0,-1}\cos(\frac{np}{N-1})$
zf-zf-zf-zf	$A_{0,0}+2A_{-1,0}\cos(\frac{p}{M}m)+4A_{1,1}\cos(\frac{p}{M}m)\cos(\frac{np}{N})+2A_{0,-1}\cos(\frac{np}{N})$
r-r-r-r	$A_{0,0}+2A_{-1,0}\cos(\frac{2p}{M}m)+4A_{1,1}\cos(\frac{2p}{M}m)\cos(\frac{2np}{N})+2A_{0,-1}\cos(\frac{2np}{N})$
Qzf-qzf-z-z	$A_{0,0}+2A_{-1,0}\cos(\frac{p}{M-1}m)+4A_{1,1}\cos(\frac{p}{M-1}m)\cos(\frac{(n+1)p}{N+1})+2A_{0,-1}\cos(\frac{(n+1)p}{N+1})$
Azf-azf-z-z	$A_{0,0}+2A_{-1,0}\cos(\frac{p}{M}(m+1))+4A_{1,1}\cos(\frac{p}{M}(m+1))\cos(\frac{(n+1)p}{N+1})+2A_{0,-1}\cos(\frac{(n+1)p}{N+1})$
z-z-zf-zf	$A_{0,0}+2A_{-1,0}\cos(\frac{p}{M+1}(m+1))+4A_{1,1}\cos(\frac{p}{M+1}(m+1))\cos(\frac{np}{N})+2A_{0,-1}\cos(\frac{np}{N})$
zf-zf-azf-azf	$A_{0,0}+2A_{-1,0}\cos(\frac{p}{M}m)+4A_{1,1}\cos(\frac{p}{M}m)\cos(\frac{(n+1)p}{N})+2A_{0,-1}\cos(\frac{(n+1)p}{N})$
zf-zf-qzf-qzf	$A_{0,0}+2A_{-1,0}\cos(\frac{p}{M}m)+4A_{1,1}\cos(\frac{p}{M}m)\cos(\frac{np}{N-1})+2A_{0,-1}\cos(\frac{np}{N-1})$
qzf-qzf-azf-azf	$A_{0,0}+2A_{-1,0}\cos(\frac{p}{M-1}m)+4A_{1,1}\cos(\frac{p}{M-1}m)\cos(\frac{(n+1)p}{N})+2A_{0,-1}\cos(\frac{(n+1)p}{N})$
z-zf-azf-qzf	$A_{0,0}+2A_{-1,0}\cos(\frac{(2m+1)p}{2M+1})+4A_{1,1}\cos(\frac{(2m+1)p}{2M+1})\cos(\frac{(2n+1)p}{2N-1})+2A_{0,-1}\cos(\frac{(2n+1)p}{2N-1})$

Tabelul 3: Valori proprii pentru diferite tipuri de conditii de granita (2D)

Capitolul 9.: Comparatie între CNN-ul cu celule de ordinul II dublu cuplate si vecinatate r=1 si CNN-ul cu celule de ordinul I cu vecinatate r=2

În cele ce urmeaza se va studia influenta ordinului celulei si a template-ului asupra dinamicii CNN-ului [81] cu template simetric.

Studiul unui CNN cu celule de ordinul I cu vecinatate r=1

Ecuatia unui CNN cu celule de ordinul I si vecinatate r=1 pentru functionarea în zona liniara din jurul originii în regim autonom este data mai jos:

$$\frac{du_i(t)}{dt} = -u_i + O_{1D}(u_i) + e \qquad i = 0..M \tag{9.1.}$$

în care pentru un template simetric

A_1	A_0	A_1

valoarea proprie spatiala este:

$$K_{1D}(m) = A_0 + 2A_1 \cos(\nu\ (m)) \tag{9.2.}$$

Se face mentiunea ca tot ceea ce s-a spus despre valori proprii spatiale si functii proprii legate de anumite conditii de granita este valabil si în cazul acesta. Prin urmare nu se mai insista în ce fel s-a obtinut relatia de mai sus.
Se face schimbarea de variabila:

$$u_i = \sum_{m=0}^{M-1} \Phi_M(m,i)\hat{u}_m \qquad i = 0..M-1 \tag{9.3.}$$

Înlocuind în ecuatia CNN-ului aceasta solutie, si folosind aceeasi tehnica folosita în cazul cu CNN format din celule de ordinul II, se obtine:

$$\frac{d\hat{u}_m(t)}{dt} = \left(-1 + K_{1D}(m)\right)\hat{u}_m + \hat{e}_m \qquad i = 0..M \tag{9.4.}$$

Se face si aici observatia se poate izola o matrice de tranzitie si o excitatie particulara, formata din sursele de curent constante din arhitectura celulelor.
Curba de dispersie este data de relatia:

$$\mathrm{Re}(l\,(K_{1D}(m))) = \mathrm{Re}(K_{1D}(m)) - 1 \tag{9.5.}$$

iar valorile proprii temporale sunt reale. Se poate aplica pentru proiectarea template-urilor metoda ferestrei mentionata în capitolele anterioare.

Studiul unui CNN cu celule de ordinul I cu vecinatate r=2

Ecuatia unui CNN cu celule de ordinul I si vecinatate r=2 pentru functionarea în zona liniara din jurul originii în regim autonom este:

$$\frac{du_i(t)}{dt} = -u_i + O_{1D}^{r=2}(u_i) + e \qquad i = 0..M \tag{9.6.}$$

în care pentru un template simetric de forma

A_2	A_1	A_0	A_1	A_2

valoarea proprie spatiala este:

$$K_{1D}^{r=2}(m) = A_0^{r=2} + 2A_1^{r=2}\cos(w(m)) + 2A_2^{r=2}\cos(2w(m)) \tag{9.7.}$$

Relatia de mai sus poate fi pusa si sub forma:

$$K_{1D}^{r=2}(m) = A_0^{r=2} - 2A_2^{r=2} + 2A_1^{r=2}\cos(w(m)) + 4A_2^{r=2}\cos^2(w(m)) \tag{9.8.}$$

Se poate rescrie:

$$K_{1D}^{r=2}(m) = a\,(K_{1D}(m))^2 + bK_{1D}(m) + g \tag{9.9.}$$

de unde rezulta ca putem exprima curba de dispersie în cazul vecinatatii de ordinul 2 în felul urmator:

$$\mathrm{Re}(l\,(K_{1D}(m))) = a\,(K_{1D}(m))^2 + bK_{1D}(m) + g - 1 \tag{9.10.}$$

Ecuatia de mai sus sugereaza de fapt ca se poate exprima curba de dispersie în cazul vecinatatii de ordinul ca functie de $K_{1D}(m)$, care este o functie monotona de moduri pe intervalul 0..M-1. Se poate rationa în acest fel în legatura cu curba de dispersie cu fereastra de largime $2A_1$ pozitionata de parametrul A_0.
În continuare se va prezenta modul de lucru folosit pentru proiectarea CNN-ului de ordinul 1 si cu template cu r=2, folosind „tehnica ferestrei".

Modul de lucru este urmatorul:

- în functie de curba de dispersie dorita se aleg parametrii α, β si γ. care stabilesc forma acesteia;

- se alege fereastra (se va vedea în exemplul de mai jos o procedura concreta de lucru) fapt care fixeaza valorile parametrilor A_0 si A_1;
- se determina valorile parametrilor din template-ul de ordinul 2 folosind setul de relatii care s-au determinat prin identificarea termenilor ultimelor doua ecuatii:

$$\begin{cases} A_0^{r=2} = a\,(A_0^{\ 2} + 2A_1^{\ 2}) + bA_0 + g - 1 \\ A_1^{r=2} = 2aA_0A_1 + bA_1 \\ A_2^{r=2} = aA_1^{\ 2} \end{cases} \tag{9.11.}$$

În cazul CNN-ului realizat cu celule de ordinul 1 si template de tip r=2 curba de dispersie în functie de $K_{1D}(m)$ este o parabola. În functie de parametrii impusi, ea se poate pozitiona, poate avea maxim sau minim si se poate dilata/contracta.

Punctele semnificative de pe curba au expresiile:

- daca α<0, atunci curba prezinta un maxim. Daca α>0, curba are un minim; Valorile maximului/minimului sunt date mai jos:

$$K_{1Dextrem} = \frac{-b}{2a} \tag{9.12.}$$

- punctul de maxim/minim (punctul cu cea mai mare valoare proprie temporala) este dat de relatia

$$\mathrm{Re}(l\,(K_{1D}))\big|\mathrm{max/min} = -\frac{b^2}{4a} + g - 1 \tag{9.13.}$$

Se observa ca parametrul γ este foarte important pentru stabilirea pozitiei pe verticala a curbei de dispersie. Facem precizarea ca acesta nu are nici o legatura cu parametrul cu acelasi nume din cazul CNN-ului de ordinul II.

- punctele în care curba de dispersie intersecteaza abscisa sunt date de relatiile:

$$\begin{aligned} K_{1Dleft} &= \min\left\{\frac{1}{2a}\left(-b \pm \sqrt{b^{\ 2} - 4a\,(g - 1)}\right)\right\} \\ K_{1Dright} &= \max\left\{\frac{1}{2a}\left(-b \pm \sqrt{b^{\ 2} - 4a\,(g - 1)}\right)\right\} \end{aligned} \tag{9.14.}$$

Conditia ca axa absciselor sa se intersecteze cu curba de dispersie este ca termenul de sub radical sa fie pozitiv:

$$b^{\ 2} - 4a\,(g - 1) > 0 \tag{9.15.}$$

În acest caz se poate izola cu ajutorul ferestrei din curba de dispersie ca functie de $K_{1D}(m)$ o regiune în care valorile proprii temporale sunt pozitive, fie ca α este pozitiv sau negativ. La limita, curba de dispersie functie de $K_{1D}(m)$ intersecteaza într-un punct axa absciselor, cel de minim sau de maxim.

O curba de dispersie tipica este cea data în figura:

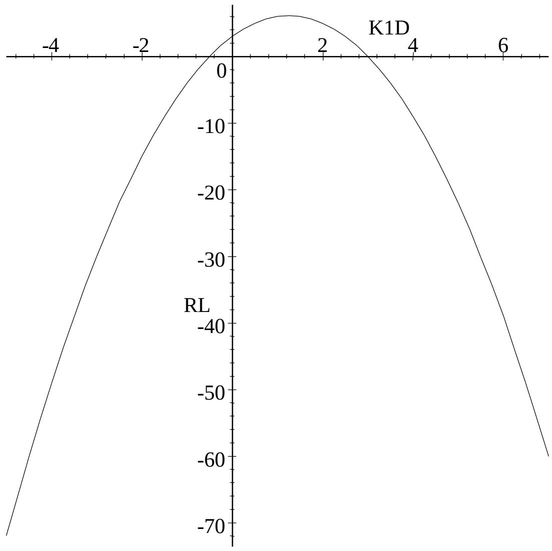


Fig. 56: Curba de dispersie pentru α=-2, β=5 γ=4

Exemplu de proiectare

Se considera un CNN cu M=30 celule cu conditii de granita de tip zero-flux. Se doreste proiectarea unui filtru trece jos banda de moduri instabile 0-m_T. Se doreste ca "amplificarea" în curent continuu sa fie unitara.

Se va fixa vârful curbei de mai sus pe axa ordonatelor. Aceasta înseamna ca parametrul β se va fixa la valoarea zero. Se noteaza cu K_T valoarea „frecventei de taiere" pentru filtrul trece jos.

Câteva restrictii pentru parametrii α, γ si pentru parametrii template-ului sunt deja cunoscute în acest moment, din forma impusa a curbei de dispersie:

$$a < 0,\; g > 1,\quad A_0 = 2A_1$$
$$K_{1Dright} = \sqrt{-\frac{g-1}{a}} = A_0 - 2A_1\cos\frac{m_T p}{M} \qquad (9.16.)$$

Din conditiile expuse, parametrii A_0 si A_1 au valorile:

$$A_0 = \frac{1}{\left(1-\cos\frac{m_T p}{M}\right)}\sqrt{-\frac{g-1}{a}}$$
$$A_1 = \frac{1}{2\left(1-\cos\frac{m_T p}{M}\right)}\sqrt{-\frac{g-1}{a}} \qquad (9.17.)$$

Pentru "amplificarea" în curent continuu dorita, (γ=2), α=-1 si m_T=10. Curba de dispersie pentru acest exemplu concret arata ca în figura:

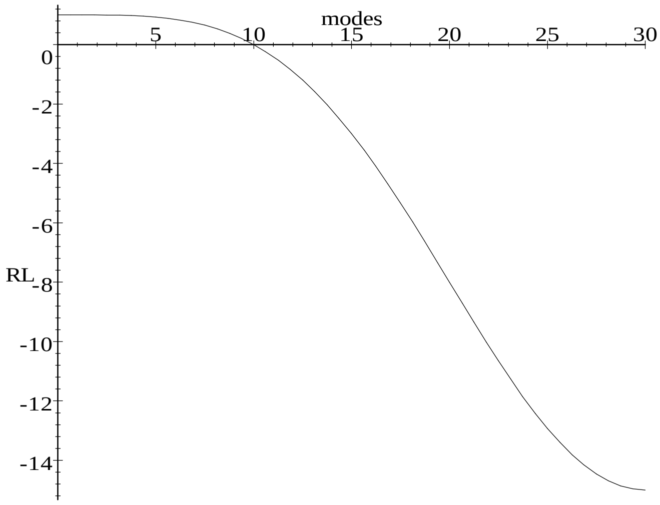


Fig. 57: Curba de dispersie pentru α=-1, β=0 γ=2, A_0=2, A_1=1

Valorile A_0=2 iar A_1=1 pozitioneaza fereastra în zona dorita si respectiv o dimensioneaza conform specificatiilor cerute . Se observa ca într-adevar "frecventa de taiere" este m_T=10 iar valoarea proprie pentru m=0 este 1, valoarea reglabila cu ajutorul lui γ. Valorile template-ului de ordinul 2 corespunzatoare sunt.

$$\begin{cases} A_0^{r=2} = -6 \\ A_1^{r=2} = -4 \\ A_2^{r=2} = -1 \end{cases} \tag{9.18.}$$

Ele sunt calculate cu ajutorul relatiei (9.12) Prin urmare, template-ul pentru problema de mai sus are forma:

-1	-4	-6	-4	-1

Si în acest caz se poate obtine dualul acestui filtru (filtru trece sus cu "frecventa de taiere" M-m_T=20 pentru acest caz:

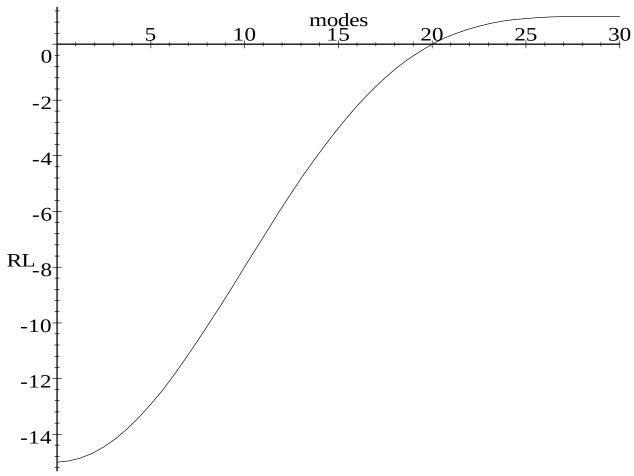


Fig. 58: Curba de dispersie pentru α=-1, β=0 γ=2, A_0=2, A_1=-1

În cele ce urmeaza vom da un exemplu de proiectare a unui filtru trece banda. Dorim ca frecventa centrala sa fie (pentru aceleasi conditii de granita si numar de celule) m_C=14 si banda sa fie Δm=10. Se pot deduce frecventele laterale:

$$m_{right} = m_C + \frac{\Delta m}{2}$$
$$m_{left} = m_C - \frac{\Delta m}{2} \tag{9.19.}$$

Valorile numerice ale acestor parametrii sunt m_{left}=9 si m_{right}=19.
Expresia benzii se calculeaza cu urmatoarea relatie:

$$\Delta K = K_{1Drigh} - K_{1Dleft} = 2\sqrt{\left(\frac{b}{2a}\right)^2 - \frac{g-1}{a}} \tag{9.20.}$$

Pe de alta parte aceeasi banda poate fi calculata (daca sunt fixati parametrii template-ului) cu relatia:

$$\Delta K = -2A_1\left(\cos\frac{m_{right}p}{M} - \cos\frac{m_{left}p}{M}\right) \tag{9.21.}$$

Expresia frecventei centrale se deduce folosind ecuatia:

$$\frac{-b}{2a} = A_0 - 2A_1 \cos\frac{m_C p}{M} \tag{9.22.}$$

Pentru ca punctul extrem sa fie unul de maxim, trebuie ca α<0. În plus, pentru ca punctul de maxim sa se afle în primul cadran, trebuie ca β>0. Un set de parametri care satisfac aceste constrângeri este:

$$\begin{cases} a = -0.5 \\ b = 4 \\ g = 6 \\ A_0 = 5.07 \\ A_1 = 5.13 \end{cases} \tag{9.23.}$$

Curba de dispersie functie de moduri este data mai jos:

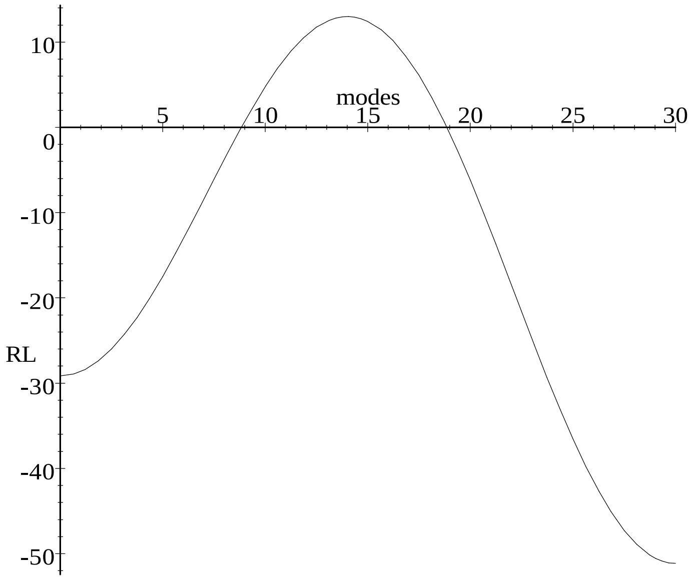


Fig. 59: Curba de dispersie pentru α=-0.5, β=4 γ=6, A_0=5.072, A_1=5.13

Se vede în mod clar banda de moduri instabile între modurile 9 si 19. Parametrii template-ului r=2 sunt dati mai jos:

$$\begin{cases} A_0^{r=2} = -14.9 \\ A_1^{r=2} = -5.5 \\ A_2^{r=2} = -13.1 \end{cases} \tag{9.24.}$$

Comparatie între CNN-ul cu celule de ordinul II dublu cuplate si vecinatate r=1 si CNN-ul cu celule de ordinul I cu vecinatate r=2

Nu se mai insista asupra rezultatelor ce descriu cantitativ dinamica CNN-ul cu celule de ordinul II dublu cuplate cu vecinatate de ordinul r=1. O curba de dispersie tipica este prezentata în Fig. 43.

Aceasta prezinta trei zone, doua laterale corespunzator carora valorile proprii temporale sunt reale si zona din mijloc, corespunzator careia valorile proprii sunt complex conjugate.

Forma curbei de dispersie în functie de K_{1D} determina comportarea celor doua tipuri de CNN, functionând ca sisteme autonome instabile. Ele pot fi comparate folosind acelasi termen (K_{1D}).

Folosind ambele tipuri de sisteme se pot construi filtre spatiale (ce filtreaza starea) de tip trece-jos, trece-sus sau trece-banda.

Daca se doreste proiectarea unui sistem cu astfel de functionalitati, este mai potrivit CNN-ul realizat din celule de ordinul I cu vecinatate de ordinul r=2.

În cazul în care se doreste însa un sistem care sa poata avea si o dinamica de tip oscilator, acest lucru se poate obtine folosind CNN-uri cu celule de ordinul II dublu cuplate si template cu raza r=1.

Capitolul 10.: Studiul dinamicii CNN-urilor simplu cuplate cu ajutorul metodei locului radacinilor si a criteriului Nyquist

Arhitectura CNN

În cele ce urmeaza se va considera o generalizare a CNN-urilor discutate pâna acum: celulele vor fi considerate în zona centrala liniara ca fiind caracterizate de o admitanta Y(s) [83, 88, 89].
De fapt, arhitectura aceasta generalizeaza arhitectura standard CNN cu celule de ordinul I, descrisa de urmatorul set de ecuatii:

$$\begin{aligned} \frac{dx_{i,j}(t)}{dt} &= -x_{i,j}(t) + \sum_{k,l\in N} A_{k,l}\, y_{i+k,j+l}(t) + \sum_{k,l\in N} B_{k,l}\, u_{i+k,j+l}(t) + I \\ y_{i,j}(x_{i,j}) &= \frac{1}{2}\left(|x_{i,j}-1| - |x_{i,j}+1|\right) \end{aligned} \qquad (10.1.)$$

În relatiile de mai sus N este dimensiunea vecinatatii celulei, i=0,1,…,M-1, j=0,1,…,N-1, A si B sunt template-uri, I este sursa de curent constant (numita si prag în terrminologia standard) iar y(x) este functia neliniara care leaga starea de iesire. Conexiunea dintre celule este data de template-ul A si este realizata cu ajutorul surselor de curent controlate în tensiune. Se utilizeaza o neliniaritate de tip saturatie. În acest caz, în partea centrala liniara a caracteristicii celulei, aceasta este descrisa de admitanta s+1-$A_{0,0}$. Aceasta observatie conduce la urmatoarea scriere a ecuatiilor ce descriu functionarea:

$$Y(s)x_{i,j}(t) = \sum_{k,l\in N} A_{k,l}\, y_{i+k,j+l}(t) + \sum_{k,l\in N} B_{k,l}\, u_{i+k,j+l}(t) \qquad (10.2.)$$

în care Y(s) este un operator integro-diferential, în particular Y(s)=s+1=d/dt+1.
În general,

$$Y(s) = \frac{Q(s)}{P(s)} \qquad (10.3.)$$

unde P(s) si Q(s) sunt polinoame în variabila s. Pentru functionarea în regiunea central liniara se pot scrie ecuatiile:

$$Y(s)x_{i,j}(t) = \sum_{k,l\in N} A_{k,l}\, x_{i+k,j+l}(t) + \sum_{k,l\in N} B_{k,l}\, u_{i+k,j+l}(t) \qquad (10.4.)$$

în care $x_{i,j}$ nu mai reprezinta în mod necesar variabile de stare ci potentialul în nodul (i,j), în timp ce $u_{i,j}$ sunt semnale de intrare, ca de obicei.

În ceea ce urmeaza vom analiza versiunea 1D a setului de ecuatii de mai sus:

$$Y(s)x_i(t) = \sum_{k \in N} A_k x_{i+k}(t) + \sum_{k \in N} B_k u_{i+k}(t) \qquad (10.5.)$$

utilizând metoda decuplarii descrisa în [60-62]. Se fac schimbarile de variabila:

$$\begin{aligned} x_i(t) &= \sum_{m=0}^{M-1} \Phi_M(m,i)\hat{x}_m(t) \\ u_i(t) &= \sum_{m=0}^{M-1} \Phi_M(m,i)\hat{u}_m(t) \end{aligned} \qquad (10.6.)$$

i=0,1,…,M-1, unde functiile $\Phi_M(\mathrm{m,i})$ sunt dependente de conditiile la limita, dupa cum s-a aratat în capitolele precedente.
Formulele de inversare sunt:

$$\begin{aligned} \hat{x}_m(t) &= \sum_{m=0}^{M-1} \Phi_M(m,i)\hat{x}_i(t) \\ \hat{u}_m(t) &= \sum_{i=0}^{M-1} \Phi_M^*(m,i)u_i(t) \end{aligned} \qquad (10.7.)$$

m=0,1,…,M-1.
Se va lucra cu functia Φ_M(m,i) de forma:

$$\Phi_M(w(m), j\,(m), i) = e^{j(w(m)i + j\,(m))} \qquad (10.8.)$$

În care ω(m) si φ(m) depind de conditiile la limita de tip periodic. Într-adevar, actiunea template-urilor A si B asupra acestor functii este urmatoarea:

$$\begin{aligned} \sum_{k \in N} A_k e^{j(w(m)i + j\,(m))} &= K_A e^{j(w(m)i + j\,(m))} \\ \sum_{k \in N} B_k e^{j(w(m)i + j\,(m))} &= K_B e^{j(w(m)i + j\,(m))} \end{aligned} \qquad (10.9.)$$

unde,

$$\begin{aligned} K_A &= A_0 + \sum_{i=1}^{N-1}(A_i + A_{-i})\cos(iw(m)) + j\sum_{i=1}^{N-1}(A_i - A_{-i})\sin(iw(m)) \\ K_B &= B_0 + \sum_{i=1}^{N-1}(B_i + B_{-i})\cos(iw(m)) + j\sum_{i=1}^{N-1}(B_i - B_{-i})\sin(iw(m)) \end{aligned} \qquad (10.10.)$$

(aici N este raza vecinatatii).

Pentru vecinatate de ordinul I, valorile proprii corespunzatoare sunt:

$$K_A = A_0 + (A_1 + A_{-1})\cos(w(m)) + j(A_1 - A_{-1})\sin(w(m))$$
$$K_B = B_0 + (B_1 + B_{-1})\cos(w(m)) + j(B_1 - B_{-1})\sin(w(m)) \tag{10.11.}$$

Ele sunt complexe pentru cazul general si reale pentru template-uri simetrice.
Cu schimbarile de variabila de mai sus, ecuatiile se decupleaza si au urmatoarea forma:

$$Y(s)\hat{x}_m(t) = K_A\hat{x}_m(t) + K_B\hat{u}_m(t) \tag{10.12.}$$

Dinamica solutiei ecuatiilor de mai sus este determinata de zerourile functiei urmatoare:

$$\frac{Q(s)}{P(s)} - K_A = 0 \tag{10.13.}$$

care poate fi rescrisa în felul urmator:

$$1 - K_A\frac{P(s)}{Q(s)} = 0 \tag{10.14.}$$

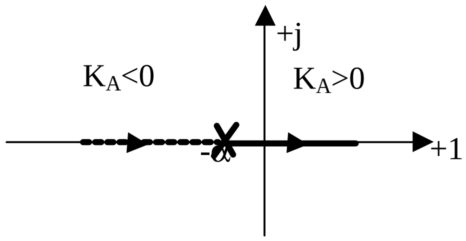
Pentru cazul CNN-urilor stabile, sisteme neautonome, se poate defini o functie de transfer pentru fiecare dintre moduri:

$$H_m(s) = \frac{K_B(m)}{\frac{Q(s)}{P(s)} - K_A(m)} \tag{10.15.}$$

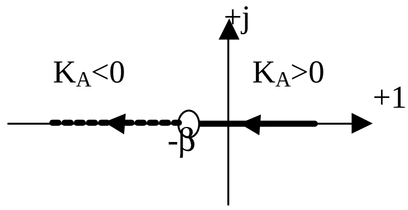
Pentru template-uri simetrice, ecuatia de mai sus poate fi tratata utilizând metoda locului radacinilor cu trasare rapida, iar valorile proprii sunt reale:

$$K_A = A_0 + 2\sum_{i=1}^{N-1} A_i\cos(iw(m)) \tag{10.16.}$$

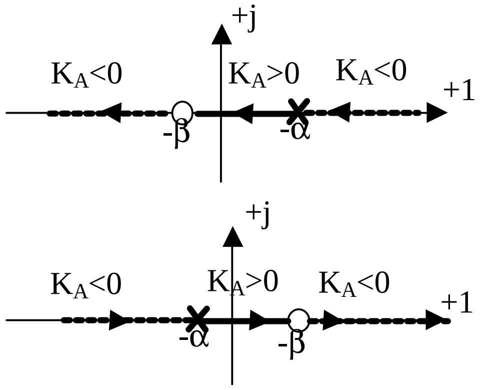
Pentru cazul general (template-uri nesimetrice) locul radacinilor se traseaza numeric.
În cele ce urmeaza se vor trasa locurile radacinilor pentru câteva forme ale admitantei Y(s)



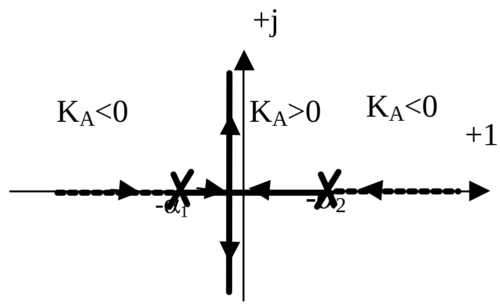
a) $Y(s)=s+\alpha$



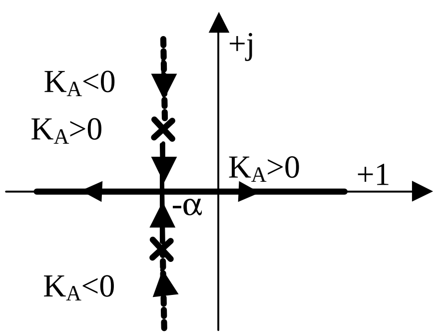
b) $Y(s)=1/(s+\beta)$



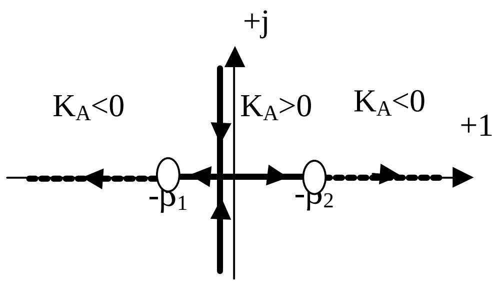
c) $Y(s)= (s+\alpha)/(s+\beta)$



d) $Y(s)= (s+\alpha 1)(s+\alpha 2)$

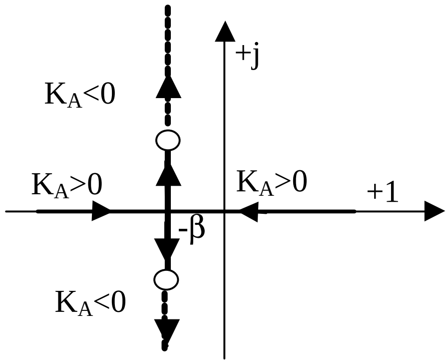


e) $Y(s)= s^2+2\alpha s+\omega_0^2$ $(w_0>\alpha)$

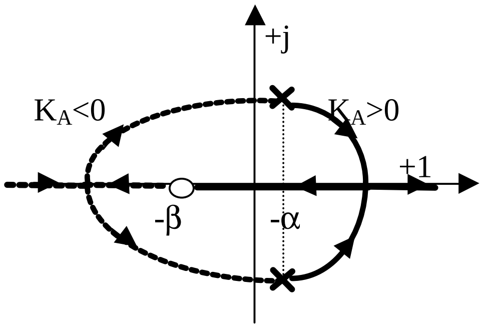


f) $Y(s)= 1/(s+\beta 1)\ (s+\beta 2)$

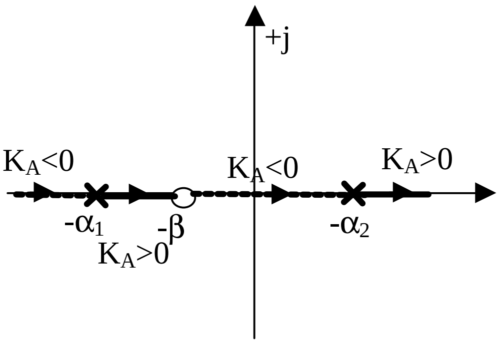
Fig. 60. Locul radacinilor pentru câteva cazuri de admitante



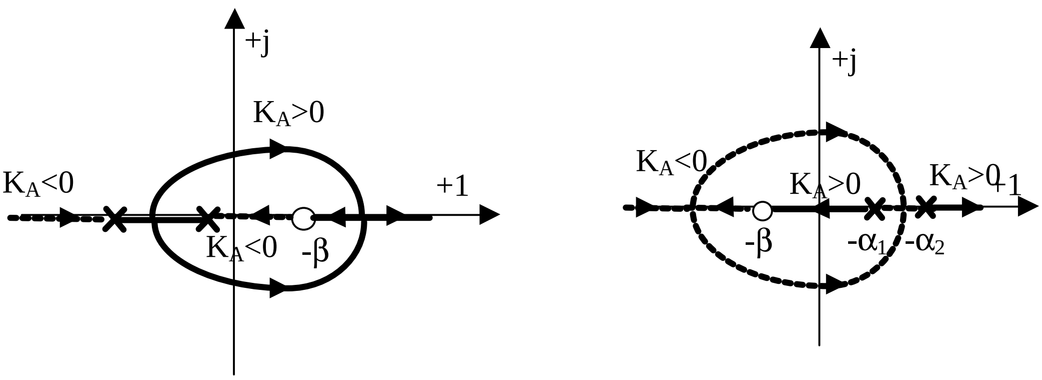
g) $Y(s)= 1/(s^2+2\beta s+\omega_0^2)$ $(w_0 > \beta)$



h) $Y(s)= (s^2+2\alpha s+\omega_0^2)/(s+\beta)$ $(w_0 > \alpha)$

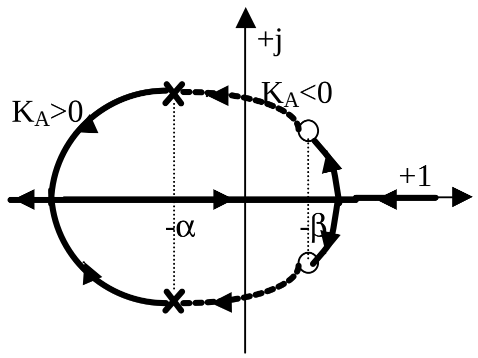


i) $Y(s)=(s+\alpha 1)(s+\alpha 2)/(s+\beta)$ în care zeroul se afla între cei doi poli

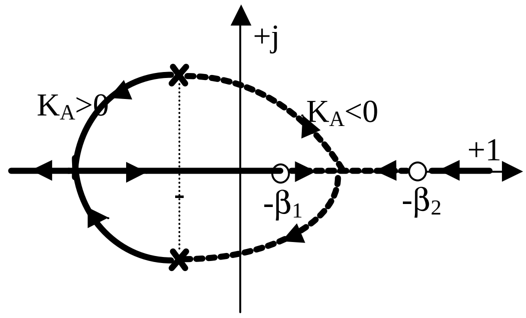


i) $Y(s)=(s+\alpha 1)(s+\alpha 2)/(s+\beta)$; când ambii poli sunt localizati la stânga zeroului sau invers radacinile pot fi complex conjugate

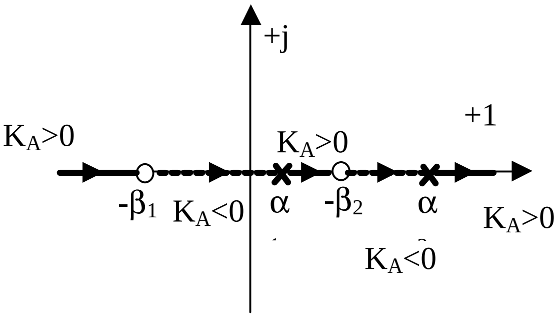
Fig.60. Locul radacinilor pentru câteva cazuri de admitante (continuare)



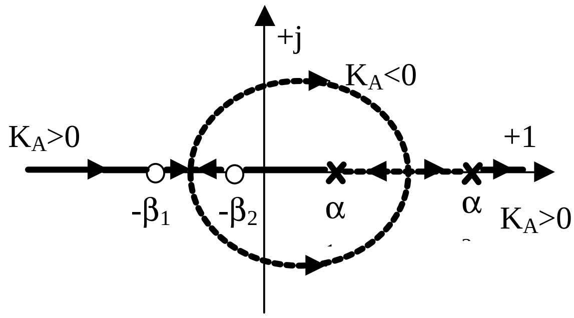
k) $Y(s)=(s^2+2\alpha s+w_1^2)/(s2+2\beta s+w_2^2)$
$(w_1>\alpha, w_2>\beta)$



l) $Y(s)=(s^2+2\alpha s+w_1^2)/(s+\beta 1)(s+\beta 2)$
$(w_1>\alpha)$



m) Y(s)=(s+α1)(s+α2)/(s+β1)(s+β2);
radacini reale pentru alternanta zero-pol



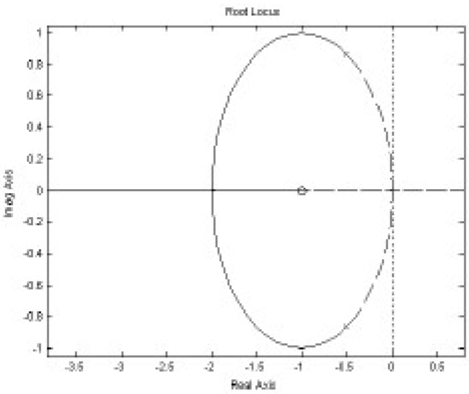
m) Y(s)=(s+α1)(s+α2)/(s+β1)(s+β2);
radacini complex conjugate pentru
zerouri si poli grupati

Fig. 60. Locul radacinilor pentru câteva cazuri de admitante (continuare)

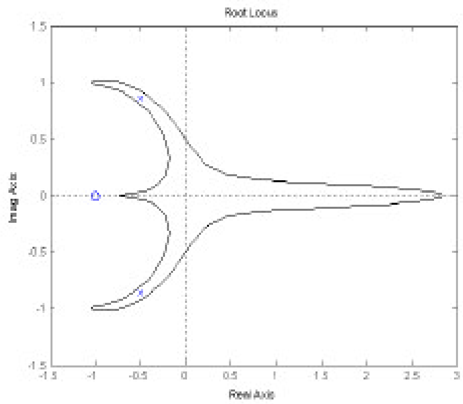
Sa notam aici ca, pentru admitante având numaratorul si numitorul interschimbate (adica polii si zerourile se interschimba), forma locului radacinilor se mentine, cu exceptia inversarii sensului sagetilor.

Trasarea LR pentru cazuri particulare ale admitantei Y(s) si template-uri asimetrice

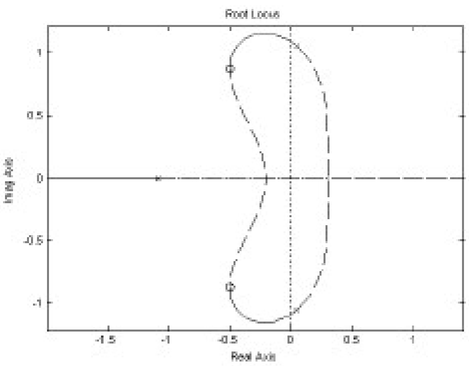
În cazul template-urilor asimetrice, forma locului radacinilor nu mai poate fi dedusa pe baza regulilor clasice. Astfel, rezultatele prezentate în continuare s-au obtinut prin simulare:



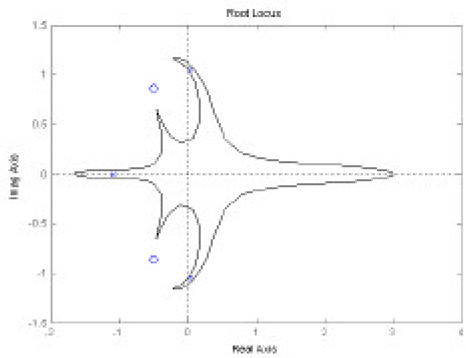
$Y(s)=(s^2+s+1)/(s+1)$, $A=[1\ 1\ 1]$



$Y(s)=(s^2+s+1)/(s+1)$, $A=[1\ 1\ 1.1]$



$Y(s)=(s^3+s^2+s+1.2)/(s^2+s+1)$, $A=[1\ 1\ 1]$



$Y(s)=(s^3+s^2+s+1.2)/(s^2+s+1)$, $A=[1\ 1\ 1.1]$

Fig. 61. Locul radacinilor pentru doua cazuri de admitante si template-uri simetrice (stânga) si asimetrice (dreapta).

Se prezinta în continuare un exemplu de loc al radacinilor pentru CNN-ul format din celule de ordinul 2 si având template de ordinul r=2.

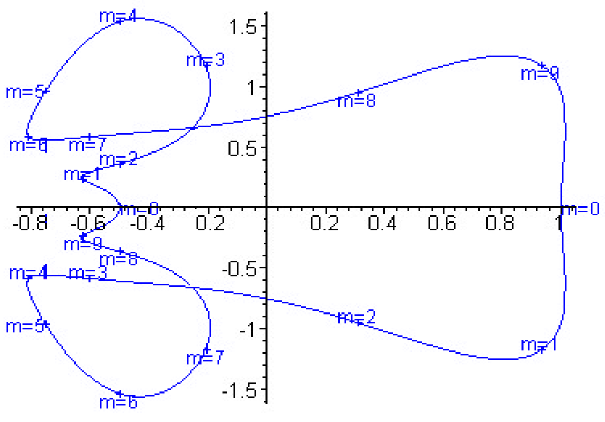


Fig. 62. Locul radacinilor pentru cazul $Y(s)=(s^2+s+1)(s+1)$ si $A=[0.5,0.5,0.5,0.5,-0.5]$

Comparatie a metodei locului radacinilor cu metoda curbei de dispersie în cazul celulei de ordinul II simplu cuplate

Metoda *curbei de dispersie* **pentru cazul CNN cu celule de ordinul II simplu cuplate, cu vecinatate de ordinul I**

În ceea ce urmeaza se vor considera celule de ordinul II simplu cuplate, functionând în zona central liniara, cuplate în vecinatati de ordinul 1 (r=1):

$$\begin{cases} \dfrac{du_i(t)}{dt} = g(f_u u_i + f_v v_i) + D_u O_{1D}(u_i) & i = 0..M-1 \\ \dfrac{dv_i(t)}{dt} = g(g_u u_i + g_v v_i) \end{cases} \tag{10.17.}$$

în care O_{1D} are forma (pentru template simetric):

$$O_{1D}(u_i) = A_{-1}u_{i-1} + A_0 u_i + A_1 u_{i+1} \tag{10.18.}$$

Se observa ca solutia sistemului (10.17) se obtine prin particularizarea solutiei sistemului de ecuatii în cazul CNN-urilor dublu cuplate pentru valoarea D_v=0. Curba de dispersie pentru CNN-uri simplu cuplate este astfel:

$$\mathrm{Re}(\lambda_{1,2}(K_{1D})) = \mathrm{Re}\left\{ g\frac{f_u + g_v}{2} + \frac{D_u K_{1D}}{2} \pm \sqrt{\left[g\frac{(g_v - f_u)}{2} - \frac{D_u K_{1D}}{2}\right]^2 + g^2 f_v g_u} \right\} \tag{10.19.}$$

Curbele din figura urmatoare reprezinta locul partii reale ale solutiilor de mai sus:

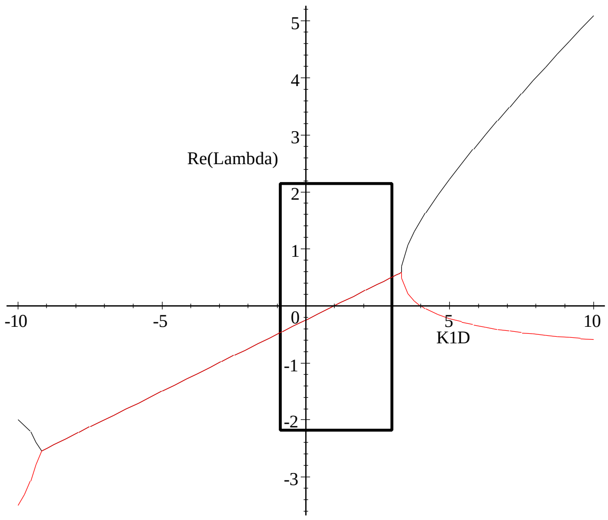


Fig. 63: Curbele de dispersie pentru CNN-uri simplu cuplate (partile reale)

Partile imaginare ale solutiilor sunt reprezentate mai jos:

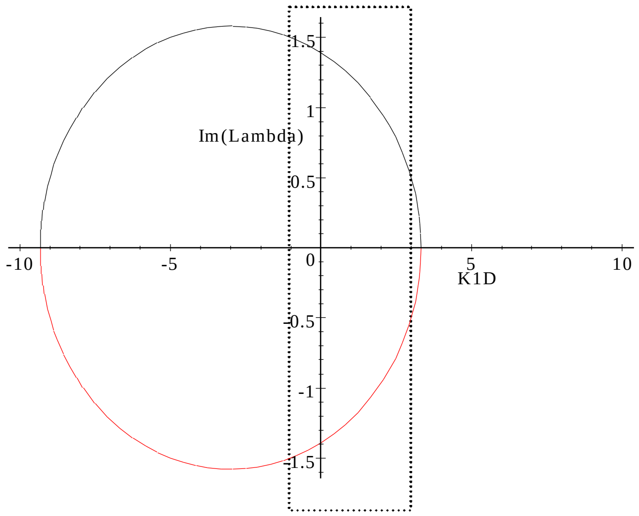


Fig. 64: Partile imaginare ale valorilor proprii

Curbele de dispersie prezinta o regiune centrala cu radacini complexe si doua regiuni laterale cu radacini reale. Judecând dupa forma curbelor si dupa domeniul în care pot lua valori valorile proprii, sunt posibile diverse dinamici. Diferenta cazului prezentat fata de cazul CNN-ului cu celule dublu cuplate este aceea ca în primul nu se poate obtine o zona centrala orizontala cu exceptia situatiei când Du=0, care corespunde celulelor neconectate. În plus, nu se pot obtine puncte de extrem pentru curba de dispersie (partea reala) în cazul CNN cuplat printr-un singur strat.

În continuare se dau câteva puncte de referinta ale curbei de dispersie pentru acest caz:

a) Extremitatile zonei în care valorile proprii sunt complex conjugate (zona de mijloc):

$$K_{1D_{s\tan ga}} = \frac{g}{D_u}\left[(f_u - g_v) - 2\sqrt{-f_v g_u}\right]$$

$$K_{1D_{dreapta}} = \frac{g}{D_u}\left[(f_u - g_v) + 2\sqrt{-f_v g_u}\right] \qquad (10.20.)$$

b) Latimea zonei centrale:

$$\Delta K_{1D} = -\frac{4g\sqrt{-f_v g_u}}{D_u} \qquad (10.21.)$$

c) Centrul zonei liniare:

$$K_{1Dmijloc} = -\frac{g}{D_u}(f_u - g_v) \qquad (10.22.)$$

si

$$\mathrm{Re}(K_{1Dmijloc}) = \frac{g\ g_v}{2} \qquad (10.23.)$$

Acestea sunt cazuri particulare ale relatiilor obtinute în cazul CNN-urilor dublu cuplate, pentru D_v=0.
Din ecuatiile precedente se poate cu usurinta vedea ca, conditia necesara pentru ca zona de mijloc sa existe ($K_{1Dstanga}$ si $K_{1Ddreapta}$ sa fie numere reale) este ca produsul $f_v g_u$ sa aiba semn negativ. Când $f_v g_u$=0, exista doar o valoare K_{1D} pentru care valorile proprii temporale sunt complex conjugate.
Partea imaginara a curbei de dispersie (în cazul în care exista) da frecventa oscilatiilor temporale pentru fiecare mod (frecventa spatiala).

$$\mathrm{Im}(l_{1,2}(K_{1D}(m)))=\sqrt{\left[g\frac{(g_v-f_u)}{2}-\frac{D_u K_{1D}(m)}{2}\right]^2+g^2 f_v g_u\}} \qquad (10.24.)$$

Selectarea unei regiuni din curba de dispersie (prin metoda fereastrei) este posibila prin alegerea adecvata a parametrilor template-ului.

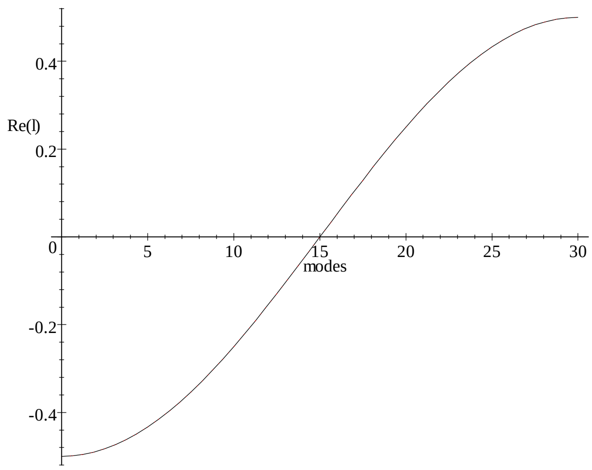


Fig. 65: Selectarea unei ferestre prin utilizarea parametrilor template-ului A_0=1 si A_1=-1 din curba de dispersie din Fig. 63 (partea reala)

Partea imaginara a acestei regiuni din curba de dispersie este reprezentata mai jos:

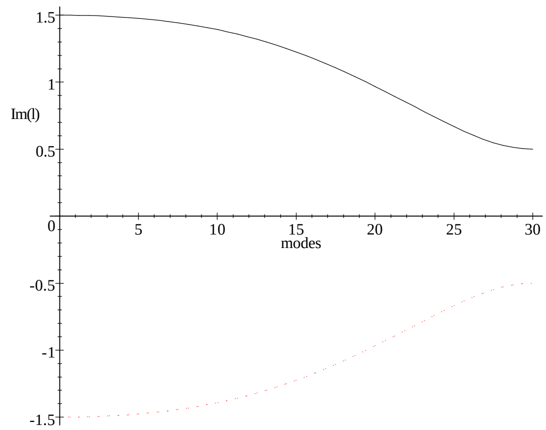


Fig. 66: Selectarea unei ferestre prin utilizarea parametrilor template-ului A_0=1 si A_1=-1 din curba de dispersie din Fig. 64 (partea imaginara)

Metoda** locului radacinilor **pentru cazul CNN cu celule de ordinul II simplu cuplate, cu vecinatate de ordinul I

Admitanta pentru descrierea celulei introduse în acest capitol pentru cazul celulei Chua [84] poate fi scrisa în forma urmatoare:

$$Y(s)=s-g\ f_u-\frac{g^2 f_v g_u}{s-g\ g_v} \tag{10.25.}$$

Când se cupleaza aceste celule într-un singur strat se obtine:

$$(Y(s)-K'_{1D}(m))\hat{u}_m \tag{10.26.}$$

Ecuatia de mai sus devine, pentru cazul CNN cu celule de ordinul II simplu cuplate, urmatoarea:

$$1-K'_{1D}\frac{s+b}{s^2+2a+v_0^2}=0 \tag{10.27.}$$

în care

$$a=-\frac{g}{2}(f_u+g_v) \tag{10.28.}$$
$$b=-g\ g_v$$
$$v_0^2=g^2(f_u g_v-f_v g_u)$$

Utilizând ecuatiile de mai sus se poate schita locul radacinilor (identificându-se cazul respectiv din cazurile prezentate în acest capitol). Mai precis, se izoleaza o regiune din curba locului radacinilor în care *K'*$K'_{1D}(m)$ ia valoare (cu ajutorul metodei ferestrei). În consecinta, punctele de pe locul radacinilor pot fi etalonate în valori de moduri.

Câteva dintre ele vor fi plasate în partea din stânga a planului complex, pe când altele se vor afla în partea dreapta a acestui plan. Punctele localizate în partea din dreapta vor corespunde modurilor instabile pe când cealalta categorie corespunde modurilor stabile.

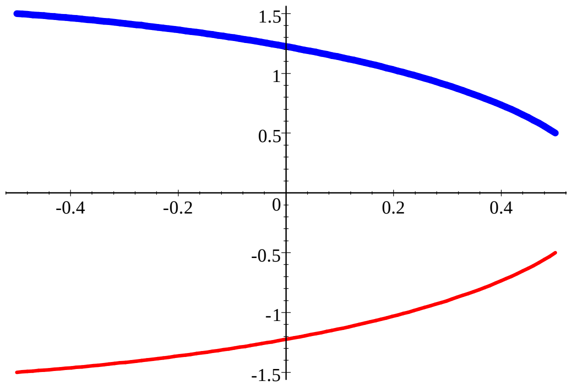


Fig. 67: Locul radacinilor când K'_{1D} este localizat în interiorul intervalului [-0.5 1.5] (echivalentul metodei ferestrei pentru metoda locului radacinilor)

Rezultatele simularii si comparatia între cele doua abordari

Simularile au fost realizate cu urmatorii parametrii: γ=5, f_u=0.1, g_u=0.1, f_v=-1, g_v=-0.2, D_u=0.5. Figurile 63 si 64 reprezinta curbele de dispersie pentru acest set de parametrii.
În Fig. 67 se prezinta locul radacinilor pentru acest set de parametrii, conform ecuatiilor (10.25) – (10.28). Dupa cum se poate vedea, zona de valori pentru partea reala a valorilor proprii în ambele reprezentari se afla între –0.5 si 0.5 (vezi Fig. 65 si 67). Partile imaginare se afla în intervalul -1.5 - 1.5 în ambele reprezentari (locul radacinilor si curba de dispersie) (vezi Fig. 66 si 67).
Conform metodei ferestrei, am izolat regiunea din curba de dispersie localizata între valorile –1 si 3 ale lui K_{1D} (vezi Fig. 63 si 64). Toate radacinile polinomului caracteristic sunt complex conjugate în acest interval.
În ceea ce priveste locul radacinilor, pentru a obtine acelasi lucru, trebuie luat în considerare ca “forta” legaturii între celule trebuie înmultita cu D_u în ecuatia (10.17).
Prin urmare, relatia între $K_{1D}(m)$ si $K'_{1D}(m)$ trebuie scrisa ca în ecuatia de mai jos:

$$K'_{1D}(m) = D_u K_{1D}(m) \tag{10.29.}$$

Ecuatia (10.29) sugereaza ca D_u nu este în mod necesar un parametru independent. Mai sunt si alti parametrii care nu sunt independenti în abordarea în care celula are o forma particulara (de exemplu, celula Chua).
Valorile echivalente pentru parametrii abordarii cu locul radacinilor sunt: α=0.25, β=1 si ω_0^2=2.
Din comparatia dintre cele doua abordari se pot deduce urmatoarele concluzii:

- metoda locului radacinilor se poate aplica si pentru CNN-uri dublu cuplate, pe când metoda locului radacinilor nu se poate aplica în acest caz;
- ceea ce se pierde la înlaturarea unui strat de legaturi este regimul pur oscilant corespunzator regiunii din centrul curbei de dispersie. De asemenea, se pierd cele doua puncte de extrem, deci posibilitatea realizarii filtrului spatial trece-banda cu acest tip de sistem;
- metoda locului radacinilor sau criteriul Nyquist se poate aplica si în cazul în care celula are orice ordin, ceea ce nu se poate spune despre metoda curbei de dispersie;
- în ambele metode nu s-a tinut seama decât de regiunea liniar centrala din functionarea celulei.

Criteriul Nyquist aplicat pentru studiul CNN-urilor simplu cuplate realizate cu celule de ordinul Q si template-uri de ordinul N

O metoda alternativa la metoda locului radacinilor pentru studiul dinamicii este aceea a criteriului lui Nyquist. Avantajul utilizarii acestei metode apare mai ales în cazul analizei dinamicii CNN-urilor cu template-uri asimetrice.
La baza analizei dinamicii sistemului folosind aceasta metoda sta observatia ca numitorul functiei de transfer pentru fiecare mod poate fi scris sub forma:

$$\frac{Q(s)}{P(s)} - K_A(m) = 0 \tag{10.30.}$$

Astfel, se reprezinta în plan complex hodograful functiei complexe Q(s)/P(s) si functia complexa $K_A(m)$. Stabilitatea sistemului este data de pozitia relativa a curbei functiei $K_A(m)$ fata de hodograful lui Q(s)/P(s). Un exemplu se da în figura de mai jos:

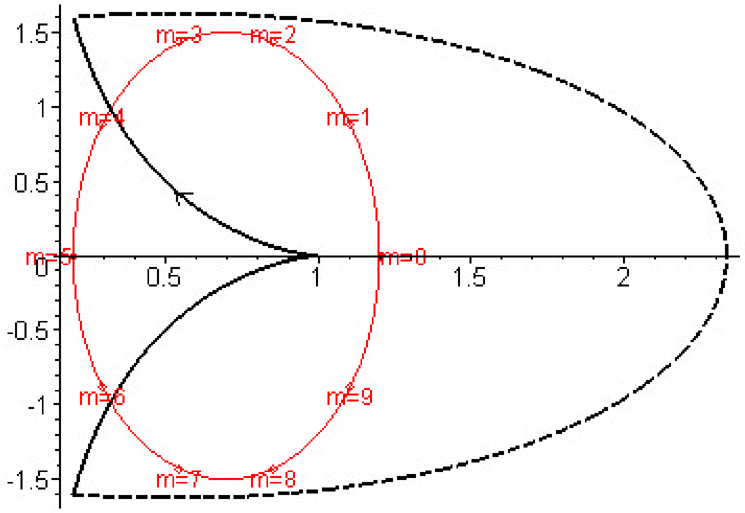


Fig. 68: Ilustrarea criteriului lui Nyquist pentru $K_A(m)$ complex
($Q(s)/P(s)=(s^2+s+1)/(s+1)$;A=[-0.5,0.7,1])

Câteva consideratii privind sinteza de „filtre pieptene”

Pentru CNN-ul realizat cu ajutorul celulelor de ordin 1 si template de ordinul N, se poate face observatia ca alegerea elementelor template-urilor pentru o anumita dinamica dorita se poate face mai usor, având în vedere observatia ca functiile de transfer pentru fiecare mod au forma:

$$H_m(s) = \frac{K_B(m)}{\frac{Q(s)}{P(s)} - K_A(m)} \qquad (10.31.)$$

Pentru cazul celulei de ordinul 1, Q(s)/P(s)=s. Prin urmare, radacinile polinomului caracteristic (valorile proprii temporale) vor fi egale cu cele spatiale:

$$l_m = K_A(m) \qquad (10.32.)$$

Când template-ul este complet (toate celulele vecine influenteaza celula curenta) coeficientii template-ului si valorile proprii temporale ale sistemului sunt perechi Fourier.

$$l_m = \sum_{k=-N}^{N} A_k e^{j\frac{2p}{M}mk} \qquad (10.33.)$$

Daca se lucreaza cu o dimensiune a template-ului mai mica (M > 2N+1) atunci este ca si cum template-ul se înmulteste cu o fereastra de "largime" 2N+1 formata numai din valori unitare (îl lasa neschimbat).
Se poate face astfel "asezarea" radacinilor polinomului caracteristic dupa cum se doreste, se calculeaza transformata Fourier inversa a acestora obtinându-se template-ul complet. Daca radacinile polinomului caracteristic au fost alese sa fie complex conjugate, template-ul rezultat va fi real.
Pentru a se trece la un template de dimensiune mai mica se înmulteste template-ul rezultat din calculul transformatei Fourier inverse cu fereastra de latime 2N+1 centrata pe elementul cu indice 0 al template-ului. În domeniul transformat, conform figurii de mai jos, înmultirii din domeniul spatiu îi corespunde produsul de convolutie. Se aseaza astfel, pentru a se obtine radacinile polinomului caracteristic corespunzator template-ului redus "sinusul atenuat" rezultat în dreptul fiecarei pozitii în care exista radacina a polinomului caracteristic si se înmulteste cu spectrul complex al template-ului neredus.
Mecanismul este exemplificat în figura de mai jos:

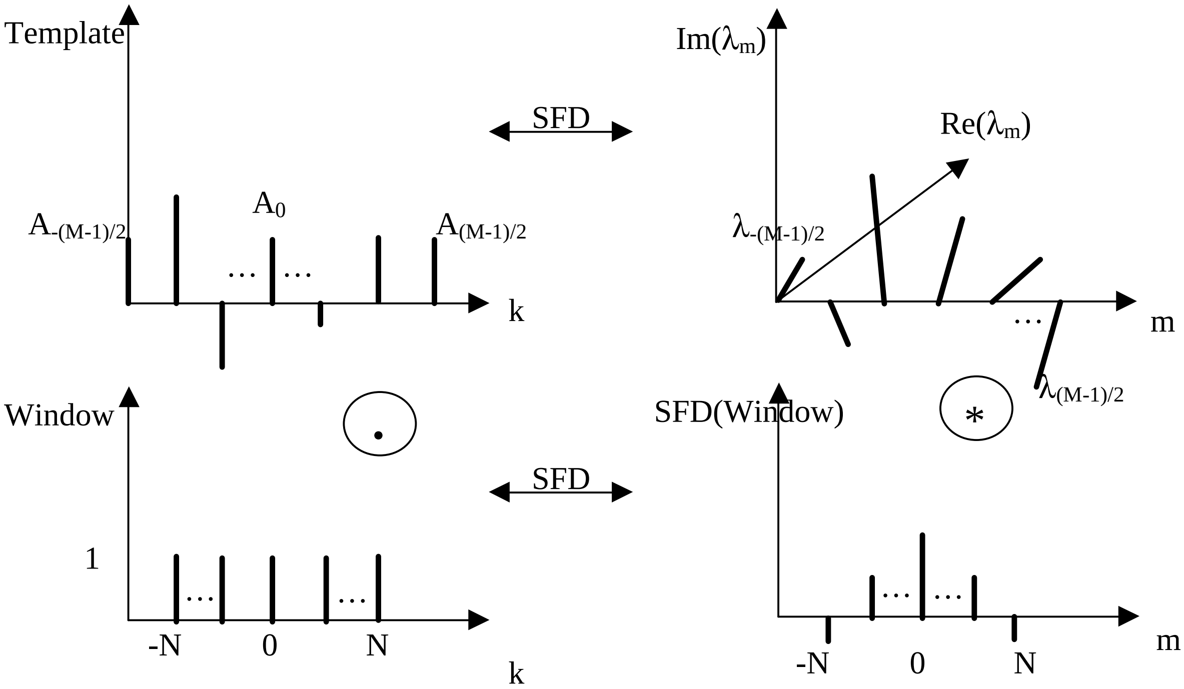


Fig. 69: Ilustrarea corespondentei template-valori proprii temporale pentru CNN cu celule de ordinul I si template de ordin r=N

De exemplu, daca initial se aleg toate radacinile egale cu zero cu exceptia uneia reala corespunzatoare lui m=0 atunci calculând inversul seriei Fourier, se obtine un template real si simetric cu toate elementele reale si egale. Produsul dintre acest template si fereastra de latime 2N+1 se transforma în domeniul frecventa în convolutie, asezând în dreptul valorii proprii cu valoarea m=0 seria Fourier corespunzatoare ferestrei.

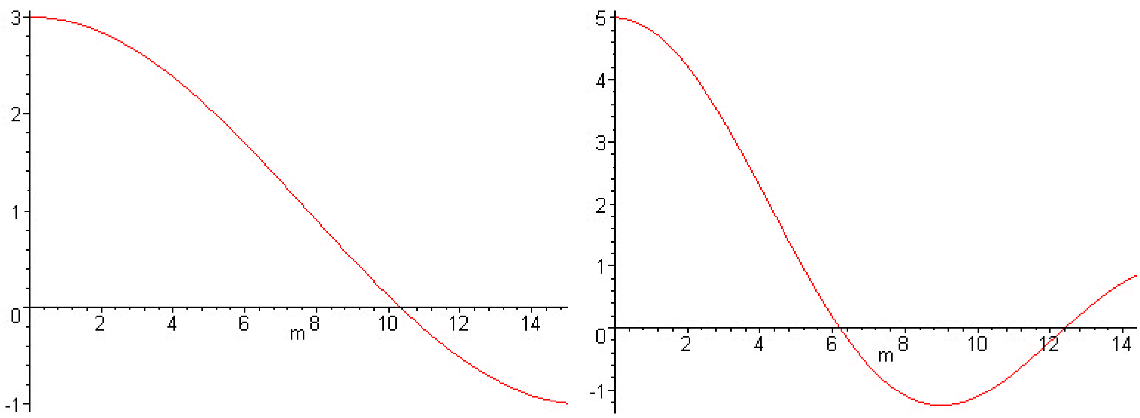
În acest caz este valabila ecuatia care da valorile proprii ale sistemului:

$$l_m = \frac{\sin\left(\frac{mp(2N+1)}{M}\right)}{\sin\left(\frac{mp}{M}\right)} \qquad (10.34.)$$

Modurile pentru care ?$_m$>0, se vor dezvolta, iar cele pentru care ?$_m$<0 vor fi "atenuate" de catre sistem.
Prin urmare, alternativ, între radacinile ecuatiei ?$_m$=0 se vor afla zone în care ?$_m$ îsi schimba semnul. Situatiile în care numaratorul valorilor proprii temporale se anuleaza sunt cele pentru care:

$$\frac{mp(2N+1)}{M} = kp \qquad (10.35.)$$

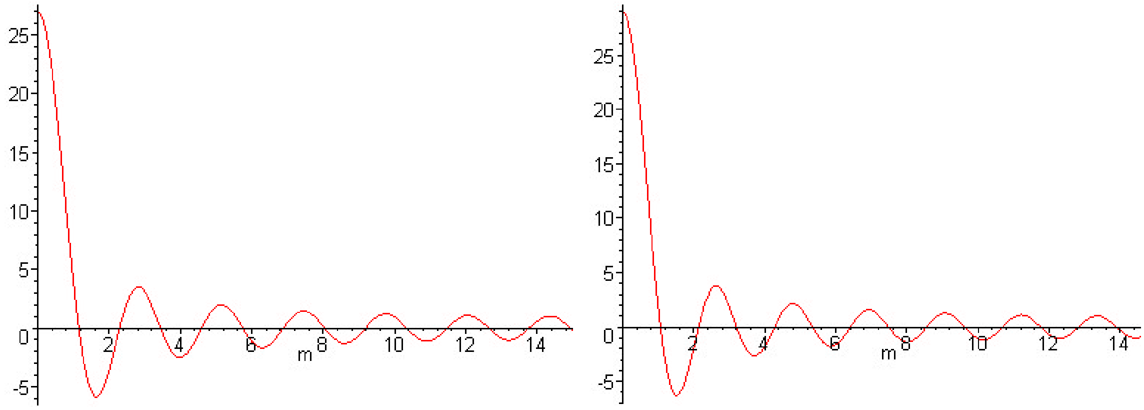
pentru k intreg.
Se întrevede astfel posibilitatea de a proiecta un CNN (cu un numar de celule prestabilit) ce are o comportare de tip "filtru pieptene" alegând adecvat dimensiunea template-ului N.
În continuare se dau câteva forme pentru curba de dispersie, în functie de dimensiunea template-ului considerat. S-a considerat un CNN cu 31 celule.



m=0	3	m=8	0.898702
m=1	2.95906	m=9	0.498695
m=2	2.837916	m=10	0.119212
m=3	2.641527	m=11	-0.22421
m=4	2.377934	m=12	-0.51752
m=5	2.057928	m=13	-0.74869
m=6	1.694611	m=14	-0.90828
m=7	1.302856	m=15	-0.98974

m=0	5	m=8	-1.09104
m=1	4.796976	m=9	-1.25
m=2	4.215849	m=10	-1.105
m=3	3.336137	m=11	-0.72552
m=4	2.276636	m=12	-0.21466
m=5	1.17714	m=13	0.309235
m=6	0.177094	m=14	0.733248
m=7	-0.60542	m=15	0.969321

Fig. 70 Curba de dispersie pentru vecinatate r=1 si respectiv r=2



m=0	27	m=8	0.139582
m=1	3.898017	m=9	-0.6137
m=2	-3.60059	M=10	0.931811
m=3	3.13249	M=11	-1.07827
m=4	-2.53253	M=12	1.054083
m=5	1.849992	M=13	-0.87663
m=6	-1.14022	M=14	0.577933
m=7	0.459588	M=15	-0.20156

m=0	29	m=8	-1.37793
m=1	1.989739	m=9	1.224212
m=2	-1.95906	m=10	-1.05793
m=3	1.908279	m=11	0.880788
m=4	-1.83792	m=12	-0.69461
m=5	1.748693	m=13	0.501305
m=6	-1.64153	m=14	-0.30286
m=7	1.517516	m=15	0.101298

Fig. 71 Curba de dispersie pentru vecinatate r=13 si respectiv r=14

Valorile proprii prezentate mai sus sunt valabile în cazul în care radacina initiala reala a polinomului caracteristic corespunzator sistemului cu template complet era egala cu 1. Daca radacina îsi schimba semnul, atunci graficele de mai sus se inverseaza. Se dau exemplele pentru r=2 si r=14 mai jos, pentru o radacina reala egala cu –1:

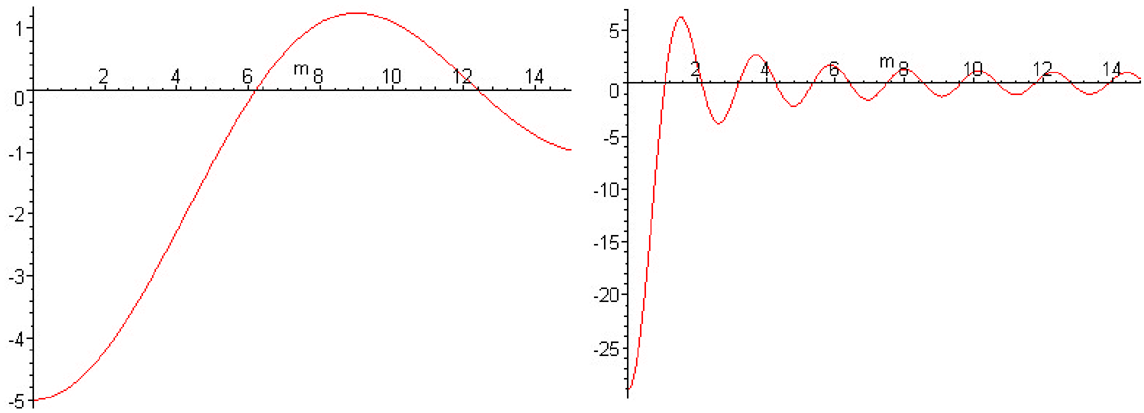
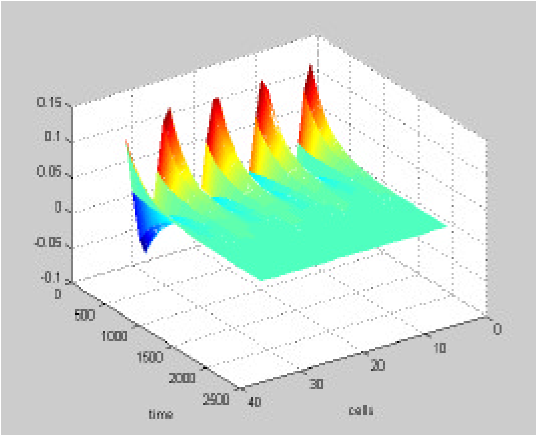
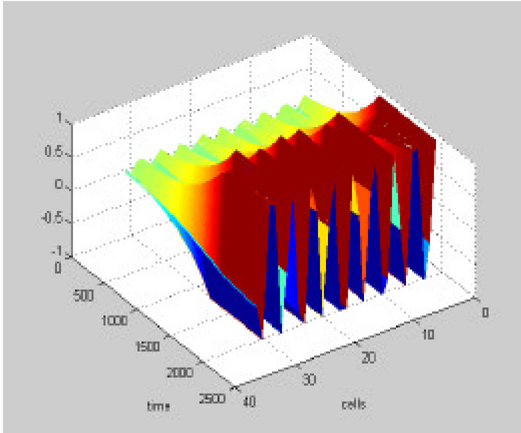


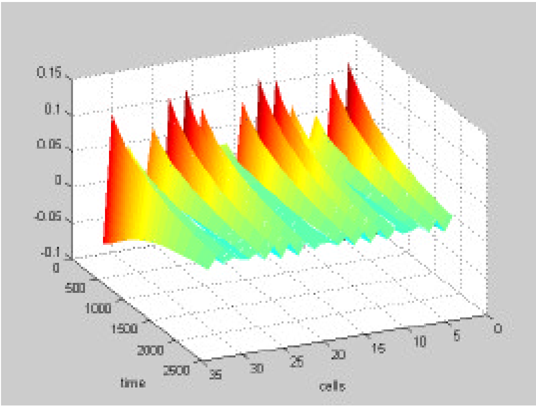
Fig. 72 Curba de dispersie pentru vecinatate r=2 si respectiv r=14 (radacina initiala reala egala cu –1)



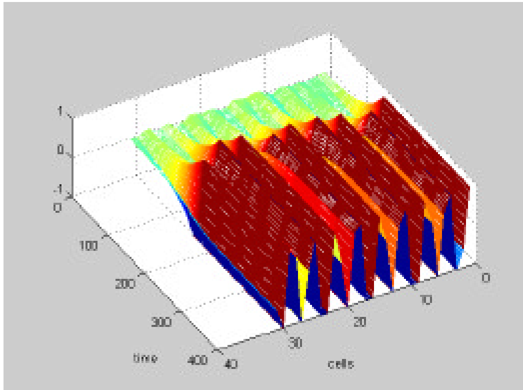
Initializare cu modul 4



Initializare cu modul 8

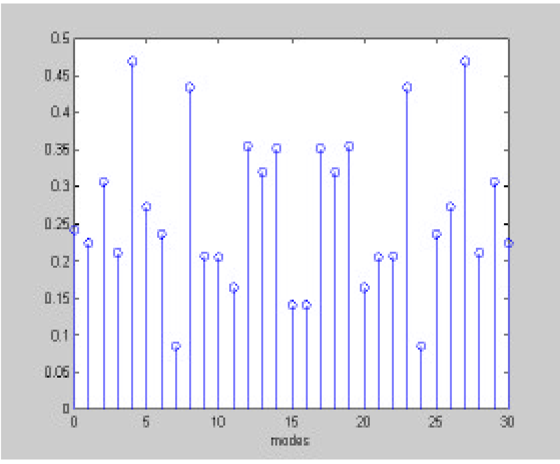


Initializare cu modul 14

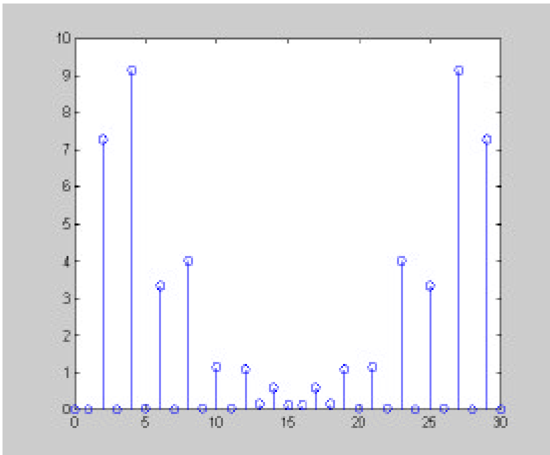


Initializare cu conditii initiale aleatoare

Fig. 73 :Rezultatele simularii sistemului cu diferite tipuri de conditii initiale pentru r=2 si radacina initiala reala egala cu –1. A=[-0.0323 -0.0323 -0.0323 -0.0323 -0.0323]



Initializare cu conditii aleatoare (spectru)



Spectru semnal spatial înainte de intrarea în neliniaritate

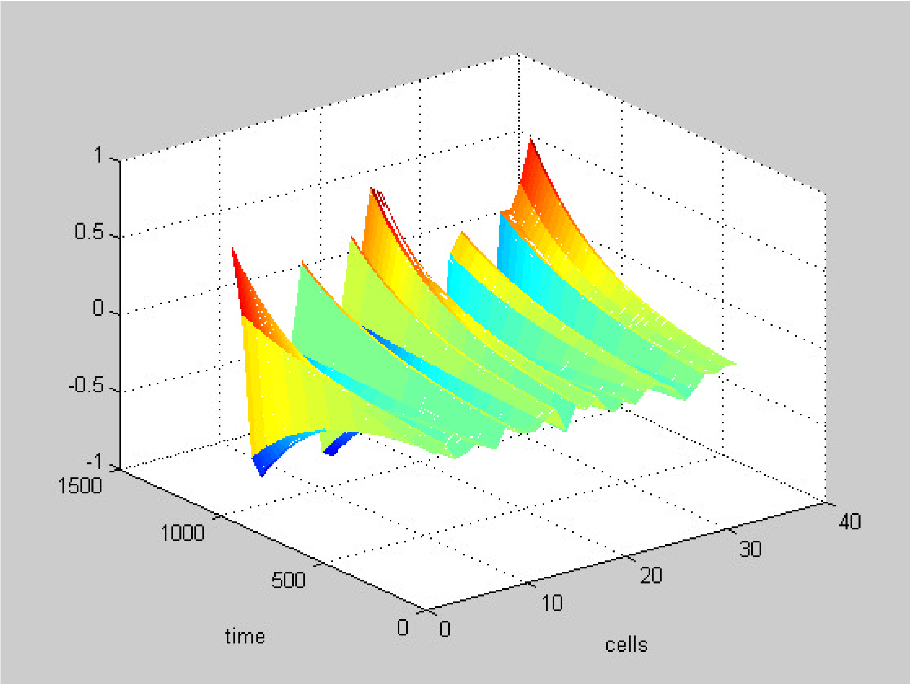


Fig. 74 :Rezultatele simularii sistemului pentru r=14 si radacina initiala[1] reala egala cu –1. A=[-0.0323 -0.0323 …. 0.9677… -0.0323 -0.0323]

Se observa ca initial starea a fost initializata cu un semnal aleator si din acesta au ramas în urma evolutiei sistemului doar modurile 2, 4, 6, 8, 10, 12, si 14, fapt ce concorda curbei de dispersie corespunzatoare.

[1] Pentru template "complet"

Capitolul 11.: Consideratii cu privire la simularea sistemelor autonome de tip CNN realizate cu celule de ordin Q si templațe de ordin N

În acest capitol se va prezenta modul în care a fost construit un simulator pentru CNN-uri din clasa mai sus mentionata. Simularea acestui tip de sisteme se bazeaza pe algoritmul Euler cu pas constant, în conditiile teoremei de existenta si unicitate a solutiei.

Descriere matematica pentru CNN-urile 1D

Ecuatia de functionare a celulei din pozitia i este:

$$Y(s)x_i(t) = \sum_{k\in N} A_k y_{i+k}(t) \tag{11.1.}$$

în care Y(s) are expresia generala:

$$Y(s) = \frac{q_Q s^Q + q_{Q-1}s^{Q-1} + \ldots + q_1 s + q_0}{p_{Q-1}s^{Q-1} + p_{Q-2}s^{Q-2}\ldots + p_1 s + p_0} \tag{11.2.}$$

Prin urmare, ecuatia de functionare a celulei de la pozitia i din retea va avea urmatoarea forma:

$$\begin{aligned} & q_Q \frac{dx_i^Q(t)}{dt^Q} + q_{Q-1}\frac{dx_i^{Q-1}(t)}{dt^{Q-1}} + \ldots + q_1 \frac{dx_i(t)}{dt} + q_0 x_i(t) = \\ & = \sum_{k\in N} A_k \left(p_{Q-1}\frac{dy_{i+k}^{Q-1}(t)}{dt^{Q-1}} + p_{Q-2}\frac{dy_{i+k}^{Q-2}(t)}{dt^{Q-2}} + \ldots + p_1 \frac{dy_{i+k}(t)}{dt} + p_0 y_{i+k}(t) \right) \end{aligned} \tag{11.3.}$$

Se fac urmatoarele notatii:

$$\begin{cases} x_0^i(t) = x_i(t) \\ x_1^i(t) = \dfrac{dx_i(t)}{dt} = \dot{x}_0^i(t) \\ \ldots \\ x_{Q-2}^i(t) = \dfrac{d^{Q-2}x_i(t)}{dt^{Q-2}} = \dot{x}_{Q-3}^i(t) \\ x_{Q-1}^i(t) = \dfrac{d^{Q-1}x_i(t)}{dt^{Q-1}} = \dot{x}_{Q-2}^i(t) \end{cases} \tag{11.4.}$$

Cu aceasta, ecuatia de mai sus se poate scrie în forma echivalenta:

$$\begin{cases} \dot{x}_0^i(t) = x_1^i(t) \\ \dot{x}_1^i(t) = x_2^i(t) \\ \dots \\ \dot{x}_{Q-2}^i(t) = x_{Q-1}^i(t) \\ \dot{x}_{Q-1}^i(t) = \dfrac{1}{q_Q}\left\{RHT^i - q_{Q-1}x_{Q-1}^i(t) - \dots - q_2x_2^i(t) - q_1x_1^i(t) - q_0x_0^i(t)\right\} \end{cases} \tag{11.5.}$$

în care

$$RHT^i = \begin{cases} \sum\limits_{k\in N} A_k \begin{pmatrix} p_{Q-1}x_{Q-1}^{i+k}(t) + p_{Q-2}x_{Q-2}^{i+k}(t) + \dots \\ + p_1x_1^{i+k}(t) + p_0x_0^{i+k}(t) \end{pmatrix}, & pt.\ \left|x_0^{i+k}(t)\right| < 1 \\ \sum\limits_{k\in N} A_k \begin{pmatrix} p_{Q-1}x_{Q-1}^{i+k}(t) + p_{Q-2}x_{Q-2}^{i+k}(t) + \dots \\ + p_1x_1^{i+k}(t) + p_0y(x_0^{i+k}(t)) \end{pmatrix}, & pt.\ \left|x_0^{i+k}(t)\right| > 1 \end{cases} \tag{11.6.}$$

Pentru scopul simularilor efectuate s-a folosit neliniaritatea clasica a CNN-ului (de tip amplificare cu saturatie). Se pot folosi orice alte tipuri de neliniaritati (functii de activare) pentru "complicarea dinamicii sistemului", în vederea obtinerii de procesari de semnal utile.

Evident, pentru a se realiza simularea sistemului evolutia variabilelor de stare ale acestuia trebuie discretizata în timp. Se obtine astfel varianta discreta în timp a sistemului respectiv, descris de ecuatiile de mai jos:

$$\begin{cases} x_0^i[n+1] = x_0^i[n] + hx_1^i[n] \\ x_1^i[n+1] = x_1^i[n] + hx_2^i[n] \\ \dots \\ x_{Q-2}^i[n+1] = x_{Q-2}^i[n] + hx_{Q-1}^i[n] \\ x_{Q-1}^i[n+1] = x_{Q-1}^i[n] + \dfrac{h}{q_Q}\left\{RHT^i - q_{Q-1}x_{Q-1}^i[n] - \dots - q_2x_2^i[n] - q_1x_1^i[n] - q_0x_0^i[n]\right\} \end{cases} \tag{11.7.}$$

Cu RHT^i având aceeasi forma ca mai sus cu exceptia trecerii argumentului t în n, corespunzator discretizarii axei timpului.
Matricial, sistemul de mai sus devine:

$$[X]^i[n+1] = [X]^i[n] + h[X_d]^i[n] \tag{11.8.}$$

în care $$[X]^i[n] = [x_0^i[n] \ x_1^i[n] \ \ldots \ x_{Q-1}^i[n]]^t \tag{11.9.}$$

iar

$$[X_d]^i[n] = \begin{cases} x_1^i[n] \\ x_2^i[n] \\ \ldots \\ x_{Q-1}^i[n] \\ \dfrac{1}{q_Q}\left(RHT^i[n] - \sum_{k=0}^{Q-1} q_k x_k^i[n]\right) \end{cases} \tag{11.10.}$$

Concluzii

În teza de fata s-au studiat diverse arhitecturi de retele neuronale, din punct de vedere al dinamicilor posibile si al stabilitatii. S-a insistat pe raportul dintre ordinul celulei (complexitatea celulei) si ordinul legaturii între celule, punându-se în evidenta prin comparatii avantajele si respectiv dezavantajele unui anume tip de CNN sau al altuia.

În continuare se va prezenta pe scurt problematica abordata în teza, punctându-se contributiile originale.

Definirea notiunilor de lucru:

- s-a început prin delimitarea notiunii de retea neuronala celulara standard si prezentarea unor clase particulare (**Cap. 1**).

Prezentarea unor rezultate importante pe studiul CNN:

- rezultate referitoare la existenta si unicitatea solutiei numerice a ecuatiilor de descriu CNN-ul si legate de marginirea solutiei în legatura directa cu implementarea la nivel de circuit a CNN-ului (**Cap. 1**).

Analiza CNN-ului stabil, functionând atât ca sistem autonom, cât si ca sistem neautonom (**Cap 2**):

- s-au prezentat solutiile ecuatiilor ce descriu CNN-ul cu celula de ordinul 1, cu template de ordin (r) maxim 2, punându-se accentul pe aplicatiile ce vizeaza CNN-urile stabile, ca filtrarea liniara spatiala;
- s-a discutat procesarea de semnal introdus pe stare, respectiv pe intrarea sistemului, corespunzând CNN-ului autonom, respectiv neautonom.

Analiza CNN-ului functionând ca sistem neliniar realizat cu celule de ordin 1 si template de ordin 1 (r=1) (**Cap. 3**):

- s-au studiat posibilitatile pe care le ofera acesta când este realizat cu celule de ordinul 1 si template de ordinul 1 (r=1);
- s-a realizat sistematizarea tehnicilor de proiectare de template-uri robuste în vederea realizarii unor procesari de imagine simple. Prin implementarea unor algoritmi în care se folosesc aceste procesari de imagine de baza se obtin altele, mai complexe;
- s-au pus în evidenta cazurile în care este nevoie ca celulele sa fie cuplate (adica elementele din template-ul A diferite de elementul central sa fie nenule). Cu ajutorul clasei de template-uri corespunzatoare se realizeaza procesari de imagine ce implica propagarea informatiei binare din imagine în CNN;
- s-a pus de asemenea în evidenta posibilitatea de a se procesa imagini pe regiuni de interes, folosind o imagine introdusa pe stare si o alta pe intrare.

Prezentarea unor aplicatii la principiile legate de functionarea CNN ca sistem neliniar (**Cap. 4**):

- a fost realizata o exemplificare a principalelor clase de aplicatii legate de procesarea de imagine;
- s-au realizat simulari care ilustreaza principiile prezentate.

S-a realizat prezentarea pe scurt a standardului actual ce descrie limbajul de nivel înalt Alpha, folosit pentru simularea CNN-urilor functionând ca sistem neliniar:

- datorita faptului ca la ora actuala exista prototipuri ale CNN-ului si un limbaj de nivel înalt (ALPHA) care asigura interfata utilizatorului cu partea hardware, s-a prezentat pe scurt acest limbaj. Clasa implementata si folosita la ora actuala este cea a CNN-urilor standard. Din acest motiv, li s-a alocat în cadrul lucrarii de fata un spatiu apreciabil (**Cap. 5**).

În cele ce urmeaza se prezinta contributiile originale legate de CNN functionând ca sistem instabil, autonom:

Functionarea CNN-ului realizat cu ajutorul celulelor de ordinul I si II, simplu si dublu cuplate, cu vecinatate cu maxim r=2 (**Cap. 6, 7, 8**):

- s-a prezentat o arhitectura generala a celulei CNN de ordinul 2 (încadrându-se realizarile de circuit prezentate în literatura anterior în aceasta arhitectura);
- s-a studiat CNN-ul ca sistem liniarizat (în jurul originii), cu imaginile introduse pe stare;
- s-a realizat studiul sistemelor obtinute prin cuplarea în simplu si dublu strat a celulelor;
- s-a studiat comportarea CNN-urilor cu celule conectate în dublu strat pentru template-uri diferite pentru cele doua straturi;
- s-au pus în evidenta (folosind curba de dispersie) avantajele si dezavantajele folosirii unei celule de ordin 2 si template de ordinul 1 si respectiv a unei celule de ordin 1 si template de ordin 2 si posibilitatea ca ele sa fie functional echivalate (**Cap. 9**);
- s-a prezentat metodologia de echivalare si s-au facut consideratii referitoare la adecvarea unei metode sau a alteia, în functie de dinamica dorita (**Cap. 9**);
- pe baza analizei acestor tipuri de sisteme (rezolvarea sistemului de ecuatii prin metoda decuplarii variabilelor) s-a schitat si o metodologie de sinteza a lor, folosind metoda ferestrei, valabila în cazul template-urilor simetrice;
- s-a exemplificat sinteza prin proiectarea unor filtre spatiale, ce fructifica functionarea CNN-ului ca sistem instabil;
- s-au realizat simulari ale CNN-urilor cu template-uri proiectate prin metodele mai sus mentionate, verificându-se practic relatiile deduse.
- s-au sistematizat conditiile la limita posibile pentru cazul 1D si respectiv 2D, punându-se în evidenta posibilitatea existentei mai multor tipuri de conditii de granita pentru grantele, respectiv laturile CNN-ului.

<u>Generalizarea metodelor de studiu a stabilitatii pentru clasa de CNN-uri autonome, simplu cuplate, alcatuite din celule de ordin Q si vecinatate de ordin N</u>: (**Cap. 10**)

- s-a studiat CNN-ul cu celula de ordin Q si template de ordin r=N, tratându-se, la nivel de sistem celula ca pe un uniport caracterizat de o admitanta Y(s). Aceasta abordare deschide orizontul de studiu al CNN-urilor simplu cuplate, cu celule legate între ele în vecinatati oricât de mari;
- s-a propus utilizarea metodei rapide de trasare a locului radacinilor, cu ajutorul careia s-a studiat stabilitatea CNN-urilor functionând în regiunea centrala liniara;
- pentru cazul template-urilor asimetrice, s-a realizat studiul CNN-urilor cu conditii de granita de tip periodic cu metoda locului radacinilor prin trasarea acestuia cu ajutorul calculatorului;
- s-a propus, ca o alternativa de studiu a stabilitatii la metoda locului radacinilor, criteriul lui Nyquist. Acesta a fost extins pentru valori proprii spatiale complexe, ce corespund template-urilor asimetrice, folosind teorema lui Cauchy;
- metodele bazate pe criteriul lui Nyquist si locul radacinilor au fost ilustrate prin simularea pe calculator a unor cazuri particulare;
- pentru CNN-rile cu celule de ordinul 2, simplu cuplate si a template-urilor simetrice s-a realizat o identificare a parametrilor de circuit ai celulei generale cu cei de sistem folositi la studiul cu ajutorul locului radacinilor, punându-se în evidenta concordanta rezultatelor obtinute prin cele doua metode.

Descrierea unei metodologii de proiectare de template-uri pentru clasa de CNN-uri autonome, simplu cuplate, alcatuite din celule de ordin 1 si vecinatate de ordin N (**Cap. 10**):

- folosind celule de ordin 1 si template-uri de ordin r=N s-a propus o metodologie de proiectare a template-urilor (simetrice sau nu), pornind de la observatia ca radacinile polinomului caracteristic si valorile template-ului complet (în care fiecare celula este legata direct cu celelalte) sunt perechi Fourier;
- s-a dat un exemplu de proiectare pentru un caz simplu, realizându-se în functie de raza de influenta filtre de tip pieptene cu un numar variabil de “dinti”;
- s-au realizat simulari ce verifica functionarea ca filtru pieptene a sistemului proiectat;
- s-a descris modul în care a fost construit simulatorul cu care s-au facut verificarile practice ale contributiilor descrise în lucrare (**Cap. 11**);

Bibliografie

[1] L.O.Chua, L. Yang – "Cellular Neural Networks: Theory", IEEE Transactions on Circuits and Systems, vol. 35, number 10, October 1988,pp 1257-1272

[2] L.O.Chua, L. Yang – "Cellular Neural Networks: Applications", IEEE Transactions on Circuits and Systems, vol. 35, number 10, October 1988,pp 1273-1288

[3] D.M.W. Leenaerts – „On Flash AD conversion”, Proceedings of IEEE International Symposium on Signals Circuits and Systems, (SCS'99), pp.37-41., Iasi - Romania, 1999

[4] L.Pivka – „Autowaves and Spatio-Temporal Chaos – Part I– A Tutorial”, IEEE Transactions on Circuits and Systems, vol. 42, number 10, October 1995,pp 638-650

[5] L.Pivka – „Autowaves and Spatio-Temporal Chaos – Part II – A Tutorial”, IEEE Transactions on Circuits and Systems, vol. 42, number 10, October 1995,pp 650-665

[6] R. Dogaru, L.O. Chua, A.T. Murgan, "Secure Communication Based on Binary Sinchronization of Chaos in Cellular Neural Networks", Proceedings of IEEE International Symposium on Signals Circuits and Systems, (SCS'97), Vol.1. 97-100, Iasi - Romania, 1997

[7] R. Dogaru and L. O. Chua, "Edge of Chaos and Local Activity Domain of FitzHugh-Nagumo Equation", Int. Journal of Bifurcation and Chaos, (Int. J. Bifurcation and Chaos), Vol. 8, No. 2, pp. 211-257, World Scientific Publishing Company, ISSN 0218-1274, 1998

[8] D.M.W. Leenaerts, "On Traveling and Solitary Waves in Modified Nagumo Equations", Proceedings of IEEE International Symposium on Signals Circuits and Systems, (SCS'97), Vol.1. pp.93-96., Iasi - Romania, 1997

[9] L. O. Chua, "A Vision of Complexity", Int. Journal of Bifurcation and Chaos, (Int. J. Bifurcation and Chaos), Vol. 7, No.10, pp. 2219-2425, World Scientific Publishing Company, ISSN 0218-1274, 1997

[10] *** - "Cellular Neural Networks", Edited by T. Roska and J. Vandewalle, J. Wiley & Sons, 1996, England"

[11] V. Cimagalli, M. Balsi – "Cellular Neural Networks: A Review", Proceedings of Sixth Italian Workshop on Parallel Architectures and Neural Networks, Vietri sul Mare, Imay 12-14, 1993, Italy

[12] L.O.Chua, Tamas Roska – "Cellular Neural Networks: Foundations and Primer", version 1.5, lecture notes for the course EE129 at U.C.Berkeley, 1997

[13] P. P. Civalleri and M. Gilli, "On Stability of Cellular Neural Networks", Journal of VLSI Signal Processing Special Issue: Spatiotemporal Signal Processing with Analogic CNN Visual Microprocessors, (JVSP Special Issue), Kluwer, 1999 November, December

[14] Leon O. Chua, "Local Activity: The Origin of Complexity", Proceedings of IEEE Int. Workshop on Cellular Neural Networks and Their Applications, (CNNA'98), pp.2, London, 1998

[15] M. Hanggi, H. C. Reddy and G. S. Moschytz, "The CNN Sampling Theorem", Proceedings of 14 European Conference on Circuit Theory and Design, (ECCTD'99), Stresa, Italy, 1999

[16] A. Zarandy, „ACE Box: High-performance Visual Computer based on the ACE4k Analogic Array Procesor Chip", Proceedings of 15'th European Conference on Circuit Theory and Design, (ECCTD'01), Espoo, Finland, 2001, pp. I-361-I-365

[17] G. Linan, R. D. Castro, S. Espejo, A.Rodriguez-Vazquez, „ACE16k: A Programmable Focal Plane Vision Processor with 128 x 128 Resolution", Proceedings of 15'th European Conference on Circuit Theory and Design, (ECCTD'01), Espoo, Finland, 2001, pp. I-345-I-349

[18] T. Roska, L.O.Chua – "The CNN Universal Machine (CNN-UM) Array Computer Architecture for Multimedia and Intelligent Sensors – Hardware and Software Design Issues – A Tutorial", Special Issue ECCTD'97, Budapest, 1997, Hungary, pp. 88-104

[19] T. Roska and L. O Chua, "The CNN Universal Machine (CNN-UM) array computer architecture for multimedia and intelligent sensors - hardware and sorfware design issues, A Tutorial", European Conference on Circuit Theory and Design - ECCTD'97, Design Automation Day proceedings, (ECCTD'97-DAD), pp.88-104, Budapest, Budapest, 1997

[20] A. Rodriguez-Vázquez, S. Espejo, R. Dominguez-Castro, "Design Considerations and Tools for the Analogic VLSI Implementation of High-Performance CNN Universal Machine", European Conference on Circuit Theory and Design - ECCTD'97, Design Automation Day proceedings, (ECCTD'97-DAD), pp. 105-110, Budapest, Budapest, 1997

[21] P. Földesy, P. Szolgay, "A CNN engine board", ECCTD-97, Design Automation Day proceedings, (ECCTD'97-DAD), pp. 199-204,

Budapest, Budapest, 1997

[22] Á. Rodriguez-Vázquez, S. Espejo, R. Dominguez-Castro, and G. Linan, "The 64x64 Analog Input CNN Universal Machine Chip and its ARAM", Proceedings of International Symposium on Nonlinear Theory and Applications, (NOLTA'98), pp. 667-670, Le Régent, Switzerland, 2-88074-391-5, 1998

[23] T. Roska, "Implementation of CNN Computing Technology", Proceedings of 7th Int. Conference Artifical Neural Networks, (ICANN'97),

pp.1151-1155, Lausanne, 1997

[24] Á. Zarándy, T. Roska, P. Szolgay, P. Földesy, S. Zöld, "A Development System for Prototyping and Interfacing CNN Chips and for Analogic Algorithm Design, Chapter 6.1.", Toward the Visual Microprocessor - VLSI Design and Use of Cellular Network Universal Machines, (Toward the Visual Microprocessor), T.Roska and A.Rodriguez-Vázquez, J.Wiley, 1997

[25] T. Roska , A. Radványi, T. Szirányi, P. Szolgay, P. Venetianer, "A CNN Application Development and Toolkit", Toward the Visual Microprocessor - VLSI Design and Use of Cellular Network Universal Machines, (Toward the Visual Microprocessor), T.Roska and A.Rodriguez-Vázquez, J.Wiley, 1997

[26] A. Paasio, A. Kananen and V. Porra, "A 176 x 144 processor binary I/O CNN-UM chip design", European Conference on Circuit Theory and Design - ECCTD'99, Design Automation Day proceedings, (ECCTD'99-DAD), Stresa, Italy, 1999

[27] Á. Zarándy, T. Roska, P. Szolgay, S. Zöld, P. Földesy and I. Petrás, "CNN Chip Prototyping and Development Systems", European Conference on Circuit Theory and Design - ECCTD'99, Design Automation Day proceedings, (ECCTD'99-DAD), Stresa, Italy, 1999

[28] K.R. Crounse, "Ph. Thesis: Image Processing Techniques for Cellular Neural Network Hardware", University of California, Berkeley, Fall, 1997

[29] K.R.Crounse, L.O. Chua – "Methods for Image Processing and Pattern Formation in Cellular Neural Networks – A Tutorial", IEEE Transactions on Circuits and Systems, Special Issue on Nonlinear Waves, Patterns and Spatio-Temporal Chaos in Dynamic Arrays, vol. 42, number 10, October 1995, pp. 583-601

[30] L.O. Chua, M. Hasler, G. S. Moschytz and J. Neirynck – "Autonomous Cellular Neural Networks: A Unified Paradigm for Pattern Formation and Active Wave Propagation",IEEE Transactions on Circuits and Systems, Special Issue on Nonlinear Waves, Patterns and Spatio-Temporal Chaos in Dynamic Arrays, vol. 42, number 10, October 1995, pp. 559-577

[31] Á. Zarándy, "The Art of CNN Template Design", Int. J. Circuit Theory and Applications - Special Issue: Theory, Design and Applications of Cellular Neural Networks: Part II: Design and Applications, (CTA Special Issue - II), Vol.17, No.1, pp.5-24, 1999

[32] A. Zarandy, T. Roska – "CNN Template Design Strategies and Fault Tolerant CNN Template Design – A Survey, Special Issue ECCTD'97, Budapest, 1997, Hungary, pp. 178-201

[33] Á. Zarándy, "The Art of binary CNN template design", Research report of the Analogic (Dual) and Neural Computing Systems Laboratory, (DNS-8-1997), Budapest, MTA SZTAKI, 1997

[34] M. Gilli, "Template Design Methodologies and Tools", European Conference on Circuit Theory and Design - ECCTD'99, Design Automation Day proceedings, (ECCTD'99-DAD), Stresa, Italy, 1999

[35] Á. Zarándy, A. Stoffels, T. Roska, and L. O. Chua, "Implementation of Binary and Gray-Scale Mathematical Morphology on the CNN Universal Machine, Paper accepted to IEEE Trans. on Circuits and Systems, 1997", IEEE Trans. on Circuits and Systems I: Fundamental Theory and Applications, (CAS-I), Vol. 45. No.2. pp. 163-168, 1998

[36] P. Földesy, L. Kék, T. Roska, Á. Zarándy, T. Roska and G. Bártfai, "Fault Tolerant Design of Analogic CNN Templates and Algorithms - Part I: The Binary Output Case", IEEE Trans. on Circuits and Systems I: Special Issue on Bio-Inspired Processors and Cellular Neural Networks for Vision, (CAS-I Special Issue), Vol. 46, No.2, pp. 312-322, 1999

[37] R. Tetzlaff, R. Kunz, G. Geis, "Analysis of Cellular Neural Networks with Parameter Deviations", Proceedings of 13 European Conference on Circuit Theory and Design, (ECCTD'97), Vol.2. pp. 650-654, Budapest, 1997

[38] J.A. Nossek, "Design and Learning with Cellular Neural Networks", Int. J. Circuit Theory and Applications, (CTA), Vol.24. issue 1. pp. 15-24., 1996,

[39] Á. Zarándy, F. Werblin, T. Roska, L.O. Chua, "Spatial Logic Algorithm using Basic Morphological Analogic CNN Operations", Int. J. Circuit Theory and Applications, (CTA), Vol 24, pp. 283-300, 1996,

[40] M. Hanggi and G. S. Moschytz, "An Exact and Direct Analytical Method for the Design of Optimally Robust CNN Templates", IEEE Trans. on Circuits and Systems I: Special Issue on Bio-Inspired Processors and Cellular Neural Networks for Vision, (CAS-I Special Issue), Vol.46, No.2, pp. 304-311, 1999

[41] M. Hanggi and G. S. Moschytz, "Making CNN Templates Optimally Robust", Proceedings of International Symposium on Nonlinear Theory and Applications, (NOLTA'98), Vol. 3, pp. 935-938, Le Régent, Switzerland, 2-88074-391-5, 1998

[42] R. Tetzlaff, R. Kunz and D. Wolf, "Minimizing the Effects of Parameter Deviations on Cellular Neural Networks", Int. J. Circuit Theory and Applications - Special Issue: Theory, Design and Applications of Cellular Neural Networks: Part II: Design and Applications, (CTA Special Issue - II), Vol.27, No.1, pp. 77-86, 1999

[43] B. Chandler, Cs. Rekeczky, Y. Nishio and A. Ushida, "Adaptive Simulated Annealing in CNN Template Learning", IEICE (Japan) Trans. on Fundamentals of Electronics, Communications and Computer Sciences, (IEICE), Vol. E82-A, No.2. Pp. 398-402, 1999

[44] B. Chandler, Cs. Rekeczky, Y. Nishio and A. Ushida, "CNN Template Learning Using Adaptive Simulated Annealing – Performance Comparison with a Genetic Algorithm", Technical Report, University of Tokushima, (TR-Tokushima), No.44, 1999

[45] A. Paasio and D. Dawidziuk, "CNN Template Robustness with Different Output Nonlinearities", Int. J. Circuit Theory and Applications - Special Issue: Theory, Design and Applications of Cellular Neural Networks: Part II: Design and Applications, (CTA Special Issue - II), Vol.27, No.1, pp. 87-102, 1999

[46] D. Liu, "Cloning Template Design of Cellular Neural Networks for Associative Memories", IEEE Trans. on Circuits and Systems I:

Fundamental Theory and Applications, (CAS-I), Vol. 44. No. 7. pp. 646-650, 1997

[47] I. Fajfar, F. Bratkovic, T. Tuma and J. Puhan, "A Rigorous Design Method for Binary Cellular Neural Networks", Int. J. Circuit Theory and Applications, (CTA), Vol. 26, No,.4, pp. 365-373, 1998

[48] M. Balsi and V. Cimagalli, "Design of CNN Filters in Lod-Polar Space", Proceedings of International Symposium on Nonlinear Theory and Applications, (NOLTA'98), pp. 675-678, Le Régent, Switzerland, 2-88074-391-5, 1998

[49] L. Nemes, L. O. Chua, T. Roska, "Implementation of Arbitrary Boolean Functions on the CNN Universal Machine", Int. J. Circuit Theory and Applications - Special Issue: Theory, Design and Applications of Cellular Neural Networks: Part I: Theory, (CTA Special Issue - I), Vol. 26. No. 6, pp. 593-610, guest ed: T. Roska, A. Rodriguez-Vazquez and J. Vandewalle, 1998

[50] F. Puffer, R. Tezlaff, D. Wolf, "A Learning Algorithm for the Dynamics with Nonlinear Templates-Part II..:Continous-Time Case",

Proceedings of IEEE Int. Workshop on Cellular Neural Networks and Their Applications, (CNNA'96), pp.467-472, Sevilla, 1996,

[51] Analogical and Neural Computing Laboratory, Hungarian Academy of Sciences – "CNN Software Library (Templates and Algorithms)", version 7.3, Budapest, 1999, Hungary

[52] Analogical and Neural Computing Laboratory, Hungarian Academy of Sciences – "CNN Alpha Language and Compiler", version 3.0G, Budapest, 2000, Hungary

[53] S. Zöld, "CNN Alpha Language and Compiler", Research report of the Analogic (Dual) and Neural Computing Systems Laboratory, (DNS-10-1997), Budapest, MTA SZTAKI, 1997

[54] P. Szolgay, K. László, L. Kék, T. Kozek, L. Nemes, I. Petrás, Cs. Rekeczky, I. Szatmári, Á. Zarándy, S. Töld and T. Roska, "The CADETWin Application Software Design System - A Tutorial", European Conference on Circuit Theory and Design - ECCTD'99, Design Automation Day proceedings, (ECCTD'99-DAD), Stresa, Italy, 1999

[55] T. Roska, P. Szolgay, T. Kozek, Á. Zarándy, Cs. Rekeczky, L. Nemes, L. Kék, K.László, I.Szatmári, M.Csapodi, "CADETWIN", Budapest, MTA SZTAKI, 1997

[56] Analogical and Neural Computing Laboratory, Hungarian Academy of Sciences – "MATCNN – Analogic CNN Simulation Toolbox for MATLAB", version 1.0, Budapest, 1998, Hungary

[57] Cs. Rekeczky, "MATCNN - Analogic CNN Simulator Toolbox for Matlab, Version 1.0", Research report of the Analogic (Dual) and Neural Computing Systems Laboratory, (DNS-11-1997), Budapest, MTA SZTAKI, 1997

[58] Analogical and Neural Computing Laboratory, Hungarian Academy of Sciences – "CadetWin'99 - System Identification: Objectives, Architecture and Components", version 3.0, Budapest, 1999, Hungary

[59] T. Roska, Á. Zarándy, S. Zöld, P. Földesy and P. Szolgay, "The Computational Infrastructure of Analogic CNN Computing - Part I: The CNN-UM

Chip Prototyping System", IEEE Trans. on Circuits and Systems I: Special Issue on Bio-Inspired Processors and Cellular Neural Networks for Vision, (CAS-I Special Issue), Vol. 46, No.2, pp. 261-268, 1999

[60] L. Goras, L.O.Chua and D.M.W. Leenaerts – "Turing Patterns in CNN's – Part I: Once Over Lightly", IEEE Transactions on Circuits and Systems, Special Issue on Nonlinear Waves, Patterns and Spatio-Temporal Chaos in Dynamic Arrays, vol. 42, number 10, October 1995, pp. 602-611

[61] L. Goras, L.O.Chua – "Turing Patterns in CNN's – Part II: Equations and Behaviors", IEEE Transactions on Circuits and Systems, Special Issue on Nonlinear Waves, Patterns and Spatio-Temporal Chaos in Dynamic Arrays, vol. 42, number 10, October 1995, pp. 612-626

[62] L. Goras, L.O.Chua, L. Pivka – "Turing Patterns in CNN's – Part III: Computer Simulation Results", IEEE Transactions on Circuits and Systems, Special Issue on Nonlinear Waves, Patterns and Spatio-Temporal Chaos in Dynamic Arrays, vol. 42, number 10, October 1995, pp. 627-637

[63] L. Goras, L. Chua, "Turing Patterns in CNN's Based on a New Cell", Proceedings of IEEE Int. Workshop on Cellular Neural Networks and Their Applications, (CNNA'96), pp.103-108, Sevilla, 1996,

[64] P. Thiran, K.R. Crounse, L.O. Chua, and M. Hasler, "Pattern Formation Properties of Autonomous Cellular Neural Networks", IEEE Trans. on Circuits and Systems I: Fundamental Theory and Applications, (CAS-I), Vol. 42. No. 10. pp. 757-774, 1995,

[65] K.R. Crounse and L. O. Chua, "Methods for Image Processing and Pattern Formation in Cellular Neural Networks: A Turorial", IEEE Trans. on Circuits and Systems I: Fundamental Theory and Applications, (CAS-I), Vol. 42. No.10. pp. 583-601, 1995,

[66] L.Goras, L.O.Chua, "On the Influence of CNN Boundary Conditions in Turing Pattern Formation", Proc. of the European Conference on Circuit Theory and Design, ECCTD'97, Budapesta, 1997, pp 383- 388;

[67] P. Arena, L. Fortuna and M. Branciforte, "Reaction-Diffusion CNN Algorithms to Generate and Control Artificial Locomotion", IEEE Trans. on Circuits and Systems I: Special Issue on Bio-Inspired Processors and Cellular Neural Networks for Vision, (CAS-I Special Issue), Vol.46, No.2, pp. 253-260, 1999

[68] P. Arena, L. Fortuna, D. Platania, and A. Rizzo, "Spatial Disorder in CNNs Induces Spatio-Temporal Organization", Proceedings of 14 European Conference on Circuit Theory and Design, (ECCTD'99), Stresa, Italy, 1999

[69] L. Goras, **T. Teodorescu**, "On CNN Boundary Conditions in Turing Pattern Formation", Proc. of the Fifth International Workshop on Cellular, Neural Networks and Their Applications, CNNA'98, pp.112-117, London, UK, 14-17 April, 1998;

[70] R. Matei, L. Goras, "On the Discrete Simulation of 1D CNN's Proceedings of SCS'97", pp. 113-117

[71] **T. Teodorescu** and L. Goras, "On the Dynamics of Turing Pattern Formation in 1D CNN's", Proceedings of IEEE International Symposium on Signals Circuits and Systems, (SCS'97), Vol.1. pp.109-112, Iasi - Romania, 1997

[72] L. Goras and L.O. Chua, "On the Role of CNN's Initial Conditions in Turing Patterns Formation", Proceedings of IEEE International Symposium on Signals Circuits and Systems, (SCS'97), Vol.1. 105-108, Iasi - Romania, 1997

[73] L. Goras and L.O. Chua, "On the Influence of CNN Boundary Conditions in Turing Pattern Formation", Proceedings of 13 European Conference on Circuit Theory and Design, (ECCTD'97), Vol. 1. pp. 383-386, Budapest, 1997

[74] B. Mirzai and G.S. Moschytz, "The Influence of the Boundary Conditions on the Robustness of a CNN", IEEE Trans. on Circuits and Systems I: Fundamental Theory and Applications, (CAS-I), Vol. 45. No.4. pp.511-515, 1998

[75] T. Serrano-Gotarredona and A. Rodriguez-Vázquez, "On the Design of Second Order Dynamics Reaction-Diffusion CNNs", Journal of VLSI Signal Processing Special Issue: Spatiotemporal Signal Processing with Analogic CNN Visual Microprocessors, (JVSP Special Issue), Kluwer, 1999 November, December

[76] E. Roca, S. Espejo, R. Dominguez-Castro, G. Linan and A. Rodriguez-Vázquez, "A programmable imager for very high speed cellular signal processing", Journal of VLSI Signal Processing Special Issue: Spatiotemporal Signal Processing with Analogic CNN Visual Microprocessors, (JVSP Special Issue), Kluwer, 1999 November, December

[77] P. Keresztes, Á. Zarándy, T. Roska, P. Szolgay, T. Bezák, T. Hídvégi, P. Jónás, A. Katona, "An emulated digital CNN implementation", Journal of VLSI Signal Processing Special Issue: Spatiotemporal Signal Processing with Analogic CNN Visual Microprocessors, (JVSP Special Issue), Kluwer, 1999 November, December

[78] P. Arena, L. Fortuna, G. Manganaro, "A CNN Cell for Pattern Formation and Active Wave Propagation", Proceedings of 13 European Conference on Circuit Theory and Design, (ECCTD'97), pp. 371-376, Budapest, 1997

[79] P. Arena, S. Baglio, L. Fortuna and G. Manganaro, "Chua's Circuit Can Be Generated by CNN Cells", IEEE Trans. on Circuits and Systems I: Fundamental Theory and Applications, (CAS-I), Vol. 42. No.2. pp. 123-125, 1995,

[80] P. Arena, S. Bagilo, L. Fortuna, G. Manganaro, "Complexity in a Two-Layer CNN", Proceedings of IEEE Int. Workshop on Cellular Neural Networks and Their Applications, (CNNA'96), pp.127-132, Sevilla, 1996,

[81] **T. Teodorescu**, L. Goras - "Cell and Template Order Influence on CNN Behavior - A Comparative Study", ECCTD 2001, Espoo, Finland

[82] L. Goras, **T. Teodorescu** - "On the Oscillatory Behavior of Second Order Cell 1D CNN's", ECCTD 2001, Espoo, Finland

[83] L. Goras, **T. Teodorescu** - "On the Dynamics of a Class of CNN", SCS 2001, Iasi, Romania

[84] **T. Teodorescu**, L. Goras - "Two Approaches for Studying Single Coupled Second Order CNN", SCS 2001, Iasi, Romania

[85] L. Goras, **T. Teodorescu**, A. Maiorescu - "Phase Influence on Mode Competition in Turing Pattern Formation", CNNA2000, Catania, Italy

[86] **T. Teodorescu**, A. Maiorescu - "Using Different Bias Current Sources for Controlling Turing Patterns", ECCTD99, Stressa, Italy

[87] **T. Teodorescu**, A. Maiorescu - "Some image processing features of the second order Hysteretic Cellular Neural Networks", ATEE98, Bucharest, Romania

[88] L. Goras, R. Ghinea, **T. Teodorescu**, E. David – „On the Dynamics of a Class of Cellular Neural Networks”, trimisa la CNNA2002, Frankfurt, Germania

[89] L. Goras, **T. Teodorescu**, R. Ghinea, E. David – „On Pattern Formation in a Class of Cellular Neural Networks”, trimisa ICSCC2002, St. Petersburg, Rusia

CUPRINS