PACHETE DE UNDINE – DE LA TEORIE LA IMPLEMENTARE SOFTWARE

Lector univ drd. ing. CHICIOREANU TEODORA DANIELA

DPPD-SSU, Universitatea Politehnica Bucuresti

Abstract

In the past decade, wavelet analysis has grown from a mathematical curiosity into a major source of new signal processing algorithms. As the signal proprieties change over time, it is preferable to isolate different time intervals with translated windows(wavelets). Wavelet pachets are particular linear combination or superpositions of wavelets. The wavelet packet transform generalizes the discrete wavelet transform and provides a more flexible tool for the time-scale analysis of data. All of the advantages of the fast wavelet transform are retained since the wavelet basis is in the repertoire of bases available with the wavelet packet transform.
The coefficients in the linear combinations are computed by a factored or recursive algorithm, with the result that expansions in wavelet packet bases have low computational complexity.

Key words: *wavelet, wavelet packet, algorithm, signal processing*

Pachetele de undine (Wavelets Packets) sau *undine arborescente* [1] sunt combinatii liniare particulare sau suprapuneri de undine. Ele formeaza baze care retin multe dintre proprietatile de ortogonalitate, finete, si localizare a undinelor parinte. Coeficientii din combinarile liniare sunt calculati printr-un algoritm factorizat sau recursiv, care arata ca expansiunile in pachetele de undine au baze cu o complexitate de calculare scazuta.

O *analiza a pachetelor de undine discrete* este o transformare in coordonate referitor la pachete de undine, in timp ce o *transformare a pachetelor de undine discrete* reprezinta doar coordonatele in privinta unei submultimi de baze.

Fie filtrele H, G dintr-o multime ortogonala. Exista doua H' si G', posibil egale cu H si G, pentru care $H * H'$ si $G * G'$ sunt proiectii ale lui ℓ^2, iar $H * H' + G * G' = I$.

Definim urmatoarele pachete de undine cu scala fixa din $\mathbb{R}$ recursiv:

$$\psi_0 \stackrel{\text{def}}{=} H\psi_0;\ \int_{\mathbb{R}} \psi_0(t)\,dt = 1, \tag{1}$$

$$\psi_{2n} \stackrel{\text{def}}{=} H\psi_n;\ \psi_{2n}(t) = \sqrt{2}\sum_{j\in\mathbb{Z}} h(j)\psi_2(2t-j), \tag{2}$$

$$\psi_{2n+1} \stackrel{\text{def}}{=} G\psi_n;\ \psi_{2n+1}(t) = \sqrt{2}\sum_{j\in\mathbb{Z}} g(j)\psi_n(2t-j). \tag{3}$$

Functia ψ_0 este unic definita si poate fi identificata cu functia ϕ din [3]. Conditia de normalizare implica faptul ca are masa unitara totala .

Functia ψ_1 este *undina mama* asociata filtrelor H si G. Colectia acestor functii ψ_n, pentru $n = 0,1, \ldots$, formeaza ceea ce vom numi *pachetele de undine* (unitate-scala) asociate lui H si G.

Formulele recursive din ecuatiile (1)-(3) confera un aranjament natural sub forma unui arbore binar, ca in Figura 1.

Definim de asemenea *pachetele duble de undine* $\{\psi_n' : n \geq 0\}$ folosind H' si G':

$$\int_{\mathbb{R}} \psi_0'(t)\, dt = 1;\ \psi_{2n}' \overset{\text{def}}{\equiv} H'\psi_n';\ \psi_{2n+1}' \overset{\text{def}}{\equiv} G'\psi_n'. \tag{4}$$

Daca privim $a(j) \overset{\text{def}}{\equiv} \sqrt{2}\psi_n(2t+j)$ ca o secventa din j pentru (t,n) fixat, atunci putem rescrie ecuatiile (2) si (3) astfel:

$$\psi_{2n}(t+i) = \sqrt{2}\sum_{j\in\mathbb{Z}} h(2i-j)\psi_n(2t+j) = Ha(i); \tag{5}$$

$$\psi_{2n+1}(t+i) = \sqrt{2}\sum_{j\in\mathbb{Z}} g(2i-j)\psi_n(2t+j) = Ga(i) \tag{6}$$

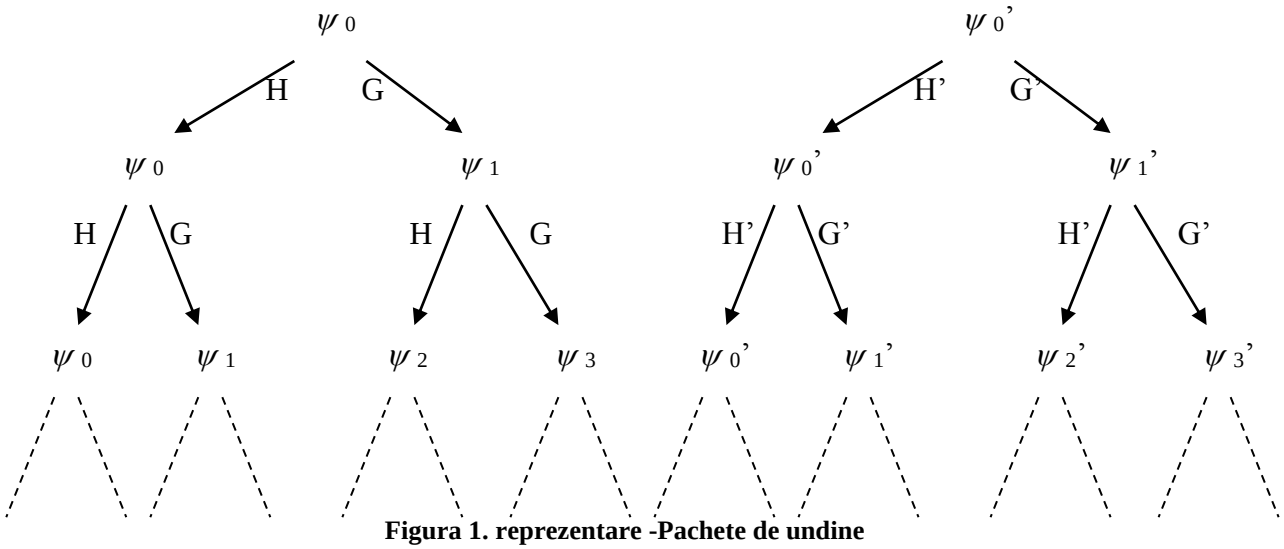

Figura 1. reprezentare -Pachete de undine

Vom nota cu Λ_n spatiul liniar inchis al translatelor integrale al pachetelor de undine unitate-scala ψ_n:

$$\Lambda_n \overset{\text{def}}{\equiv} \left\{x(t) = \sum_k c(k)\psi_n(t-k)\right\} \subset L^2(\mathbb{R}). \tag{7}$$

Din moment ce translatele se intrepatrund cu filtrele H si G, ecuatiile (1)-(3) implica faptul ca

$$\Lambda_{2n} = H\Lambda_n \text{ si } \Lambda_{2n+1} = G\Lambda_n. \tag{8}$$

Observam ca $\Lambda_0 = V_0$ spatiul de aproximare unitate-scala definit de filtrul de joasa trecere H. Avem o baza naturala pentru fiecare dintre aceste spatii:

Multimea $\{\psi_n(t-k) : k \in \mathbb{Z}\}$ *este o baza pentru* Λ_n.

Aratam ca punand bazele $\Lambda_n, n \geq 0$ impreuna rezulta baza pentru $L\hat{}2(\mathbb{R})$. Realizam asa adaptand usor si apoi translatand notatie noastre metoda lui Coifman si Meyer [4], pentru a include si cazul biortogonal. Folosind filtrele duble H' si G' si proprietatea de reconstructie exacta, obtinem:

$$\psi_n(t+j) = \frac{1}{\sqrt{2}} \sum_{i \in \mathbb{Z}} \bar{h}'(2i-j)\psi_{2n}\left(\frac{t}{2}+i\right) + \sum_{i \in \mathbb{Z}} \bar{g}'(2i-j)\psi_{2n+1}\left(\frac{t}{2}+i\right). \tag{9}$$

Astfel, daca $x = x(t) = \sum_{k \in \mathbb{Z}} \bar{\lambda}_n(k)\psi_n(t+k)$, putem folosi ecuatia (9) pentru a scrie urmatoare a pereche de siruri pentru x:

$$\begin{aligned} x(t) &= \frac{1}{\sqrt{2}} \sum_i \left(\sum_j \bar{h}'(2i-j)\bar{\lambda}_n(j) \right) \psi_{2n}\left(\frac{t}{2}+i\right) + \\ &+ \frac{1}{\sqrt{2}} \sum_i \left(\sum_j \bar{g}'(2i-j)\bar{\lambda}_n(j) \right) \psi_{2n+1}\left(\frac{t}{2}+i\right) = \\ &= \sum_i \overline{H'\lambda_n(\iota)} \frac{1}{\sqrt{2}} \psi_{2n}\left(\frac{t}{2}+i\right) + \sum_i \overline{G'\lambda_n(\iota)} \frac{1}{\sqrt{2}} \psi_{2n+1}\left(\frac{t}{2}+i\right). \end{aligned} \tag{10}$$

In partea dreapta se afla expansiunea lui x, folosind functii de dilatare in baza Λ_{2n} si Λ_{2n+1} avand coeficientii $\overline{H'\lambda_n}$ si $\overline{G'\lambda_n}$. Echivalent, pentru oricare $x \in \Lambda_n$ putem scrie

$$x(t) = \frac{1}{\sqrt{2}} y\left(\frac{t}{2}\right) + \frac{1}{\sqrt{2}} z\left(\frac{t}{2}\right), \text{pentru } y \in \Lambda_{2n} \text{ si } z \in \Lambda_{2n+1}. \tag{11}$$

Calcularea numerica a coeficientilor pachetelor de undine

Fie $\{\lambda_{sf}(p) : p \in \mathbb{Z}\}$ secventa de produse interioare ale functiei $x = x(t)$ in $L^2(\mathbb{R})$ cu functiile de baza "inversate" in $\sigma^2\Lambda_f$:

$$\lambda_{sf}(p) \stackrel{\text{def}}{=} \langle x, \psi^{<}_{sfp} \rangle = \int_{\mathbb{R}} \bar{x}(t) 2^{-s/2} \psi_f(p - 2^{-s}t)\, dt. \tag{12}$$

Aici $s, p \in \mathbb{Z}, f \geq 0$, si $\psi^{<}_{sfp}(t) \stackrel{\text{def}}{=} 2^{-s/2}\psi_f(p - 2^{-s}t)$. Prin dualitate, daca $x = \psi'^{<}_{sfp'}$, vom avea $\lambda_{sf}(p) = \delta(p - p')$. Similar, numerele $\bar{\lambda}_{sf}(p)$ sunt coeficienti ai expansiunii x in functiile $\sigma^s\Lambda'_f$:

$$x(t) = \sum_p \bar{\lambda}_{sf}(p)\psi'^{<}_{sfp}(t) \Longrightarrow \langle x, \psi^{<}_{sfp} \rangle = \lambda_{sf}(p). \tag{13}$$

Vom folosi de asemenea notatia $\{\lambda\}$ pentru $\{\lambda_{00}\}$, produsele interioare "inverse" ale lui $x(t)$ cu functiile de baza $\sigma^0\Lambda_0 = V_0$. Din acestea putem calcula produsele interne ale lui $x(t)$ cu pachetele de undine din orice spatiu $\sigma^s\Lambda_f$, pentru $s > 0$ si $0 \leq f < 2^s$, aplicand operatorii H si G pentru un numar total de s ori. Aceasta este o consecinta a urmatoarelor:

Secventele de coeficienti $\{\lambda_{sf}\}$ *satisfac relatiile*

$$\lambda_{s+1,2f}(p) = H\lambda_{sf}(p), \tag{14}$$

$$\lambda_{s+1,2f+1}(p) = G\lambda_{sf}(p). \qquad (15)$$

Demonstratie:

$$\lambda_{s+1,2f}(p) = \int_{\mathbb{R}} \bar{x}(t)2^{\frac{-s-1}{2}}\psi_{2f}(p - 2^{-s} - 1)\, dt =$$

$$= \int_{\mathbb{R}} \bar{x}(t)2^{\frac{-s-1}{2}}\left[2^{\frac{1}{2}}\sum_{j\in\mathbb{Z}} h(j)\psi_f(2[p - 2^{-s-1}t] - j)\right] dt =$$

$$= \sum_{j\in\mathbb{Z}} h(j)\int_{\mathbb{R}} \bar{x}(t)2^{\frac{-s}{2}}\psi_f(2p - j - 2^{-s}t)\, dt =$$

$$= \sum_{j\in\mathbb{Z}} h(j)\lambda_{sf}(2p - j) =$$

$$= H\lambda_{sf}(p).$$

Colectia de pachete de undine $\{\psi_{sfp}\} = \{2^{s/2}\psi_f(2^s t - p)\}$ folosite in produsele interioare cuprinde o librarie de functii, cu organizari naturale derivand din (14-15). Le putem grupa sub forma de arbore binar ale carui noduri sunt spatiile $\sigma^2\Lambda_f$. Radacina este $V_0 = \Lambda_0$, frunzele sunt $\sigma^L\Lambda_0, \ldots, \sigma^L\Lambda_{2^L-1}$, iar genealogia urmeaza diagrama din partea dreapta a Figurii 2. Fiecare nod este suma celor doi descendenti imediat, sau *copii*. Daca H,G sunt ortogonale, suma este o suma direct ortogonala. De asemenea, daca pornim cu o secventa de coeficienti ai pachetelor de undine unitate-scala $\lambda = \lambda_{00}$, atunci secventele λ_{sf} de coeficienti ai pachetului de undine multiscala formeaza nodurile arborelui binar din partea stanga a diagramei din Figura 2.

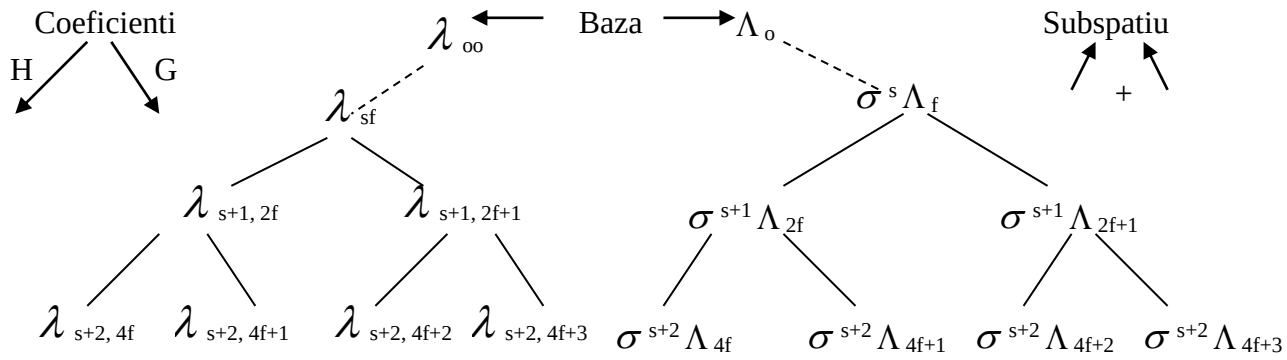

Figura 2. Analiza -pachete de undine

Pentru a obtine functia de analiza a pachetelor de undine, mai intai trebuie sa ii gasim secventa de coeficienti in subspatiul radacina, apoi urmarim ramurile arborelui coeficientilor pachetelor de undine pentru a gasi expansiunea din subspatiile descendente. Ramurile din arbore corespund indicilor s, f si astfel secventelor de filtre H si G. Corespondenta poate fi calculata folosind inductia din (14-15). Daca definim $f = (f_{k-1}\cdots f_1 f_0)_2$ ca fiind reprezentarea binara a valorii $f \in [0, 2^k - 1]$, obtinem urmatoarele:

Teorema: Pentru oricare $s \geq 0$ *si* $0 \leq f < 2^s$ *avem* $\lambda_{sf} = F_0 \cdots F_{s-1}\lambda(p)$, *unde* $F_i = H$ *daca* f_i *este* 0, *altfel* $F_i = G$.

Algoritmul inclus in aceasta teorema are complexitate scazuta in urmatorul sens. Sa presupunem ca ni se dau coeficientii pachetelor de undine unitate-scala $\{\lambda(p): p \in \mathbb{Z}\}$ ai functiei $x = x(t)$. Sa

presupunem ca $|\lambda(p)|$ este neglijabil de mic, atata timp cat conditia $|p| < N$ nu este satisfacuta. Daca folosim filtre cu suport finit, fiecare aplicare a lui H sau G asupra secventelor de coeficienti ne costa $O(N)$ Atunci Teorema ne ofera o constructie simpla pentru toate produsele interioare neneglijabile $\lambda_{sf}(p)$: complexitatea producerii tuturor este $O(sN) \approx O(N \log N)$. Aceasta margineste costul unei *analize a pachetelor de undine*, cu alte cuvinte, descoperirea coeficientilor $\{\lambda_{sf}(p)\}$ fiind dat $\{\lambda(p)\}$.

Fiecare aplicare a filtrelor adiacente cu suport finit, H'^* sau G'^* au de asemenea un cost de $O(N)$. Asadar, *sinteza pachetelor de undine*, sau reconstructia $\{\lambda_{00}\}$ dati fiind coeficientii $\{\lambda_{sf}(p)\}$, este de asemenea un algoritm de complexitate $O(sN) \approx O(N \log N)$.

Implementare

Pachetele de undine pot fi construite folosind limbajul C++/C# (necesitand avansate cunostinte de programare) sau folosind programul MATLAB (acest program ne ofera pachete de instructiuni deja create, necesitand cunostinte elementare de programare).

Costructia pachetelor de undine se poate realiza folosind recursivitatea [6] (structura de baza a pachetelor de undine este perfect recursiva):

```
packtree::packtree( const double *vec, const
size_t N,liftbase<packcontainer, double> *w )
{
 waveObj = w;
 block_pool mem_pool;
 double *vecCopy = (double
*)mem_pool.pool_alloc( N * sizeof( double )
);
 for (int i = 0; i < N; i++) {
        vecCopy[i] = vec[i];  }
 root = new packnode<double>( vecCopy, N,
packnode<double>::OriginalData );
 root->mark( true );
 newLevel( root, false, false );
} // packtree

void  packtree::cleanTree(packnode<double>
*top, bool removeMark )
{
 if (top != 0) {
  if (removeMark) {
   if (top->mark()) {
    top->mark( false );    }
                    }
  else {
   if (top->mark()) {
    removeMark = true;     }
  }
  cleanTree( top->lhsChild(), removeMark );
  cleanTree( top->rhsChild(), removeMark );
 }
} // cleanTree
void packtree::prCost()
{  if (root != 0) {
   breadthFirstPrint(printCost);  }}

void packtree::prBestBasis()
{  if (root != 0) {
    cleanTree( root, false );
    breadthFirstPrint(printBestBasis);
 }} // prBestBasis

double packtree::bestBasisWalk( packnode
<double> *top )
{  double cost = 0.0;
    if (top != 0) {
    packnode<double> *lhs = top->lhsChild();
    packnode<double> *rhs = top->rhsChild();
    if (lhs == 0 && rhs == 0) {
       cost = top->cost();    }
  else if (lhs != 0 && rhs != 0) {
  double lhsCost = bestBasisWalk( lhs );
  double rhsCost = bestBasisWalk( rhs );
  double v1 = top->cost();
  double v2 = lhsCost + rhsCost;
  if (v1 <= v2) {
     top->mark( true );
     lhs->mark( false );
     rhs->mark( false );     }
   else {  top->cost( v2 );  }
   cost = top->cost();    }
  else {    assert( false );    }  }
```

```
  return cost;
} // bestBasicWalk
void packtree::bestBasis()
{
  bestBasisWalk( root );
} // bestBasis

void packtree::checkBestBasis(
packnode<double> *top )
{
  if (top != 0) {
    if (top->mark()) {
      foundBestBasisVal = true;
      if (top->getKind() == packdata<double>::OriginalData)
           { foundOriginalData = true;      }
    }
    if (!foundOriginalData) {
      checkBestBasis( top->lhsChild() );
    }
    if (!foundOriginalData) {
      checkBestBasis( top->rhsChild() );
    }
  }
}  // checkBestBasis

void packtree::buildBestBasisList(
packnode<double> *top,    packdata_list<double>
&list )
{   if (top != 0) {
    if (top->mark()) {
      list.add( top );    }
    else {
      buildBestBasisList( top->lhsChild(), list );
      buildBestBasisList( top->rhsChild(), list );
    }  }} // buildBestBasisList

void          packtree_base::breadthFirstPrint(const
printKind kind)
{  queue<double> Q;
   Q.addQueue( root, 0 );
  while (! Q.queueEmpty() ) {
    packnode<double> *node = Q.queueStart()->node;
    size_t indent = Q.queueStart()->indent;
    Q.deleteStart();
       if (indent > 0) {
         printf("%*c", indent, ' ');
    }
    switch (kind) {
    case printData: node->pr();      break;
    case printCost: node->prCost();     break;
    case printBestBasis: node->prBestBasis();
                 break;
    default:      assert( false );
      break;
    } // switch
    packnode<double> *lhs = node->lhsChild();
    packnode<double> *rhs = node->rhsChild();
    if (lhs != 0) {
      Q.addQueue( lhs, indent + 2 );
    }
    if (rhs != 0) {
      Q.addQueue( rhs, indent + 2 );
    }
  }
} // packtree_base::breadthFirstPrint
```

Pentru analiza pachetelor de undine, programul MATLAB ne ofera urmatoare functii predefinite[7]: wpcoef , wpdec , wpdec2 , wpsplt, wprcoef, wprec , wprec2, wpjoin , besttree, bestlevt, entrupd, get, read , wenergy, wp2wtree, wpcutree, ddencmp, wpbmpen, wpdencmp, wpthcoef, wthrmngr.

Cu ajutorul acestora putem sa prelucram semnale, sa costruim pachete de undine etc.

```
load noischir; x = noischir;
n = length(x);
thr = sqrt(2*log(n*log(n)/log(2)));
xwpd = wpdencmp(x,'s',4,'sym4','sure',thr,1);
xwd = wden(x,'rigrsure','s','one',4,'sym4');
subplot(311),plot(x), xlim([1 n])
title('Semnal original')
subplot(312),plot(xwpd), xlim([1 n])
title('Semnal prelucrat folosind pachete de undine')
subplot(313),plot(xwd), xlim([1 n])
title('Semna prelucrat folosind undine')
```

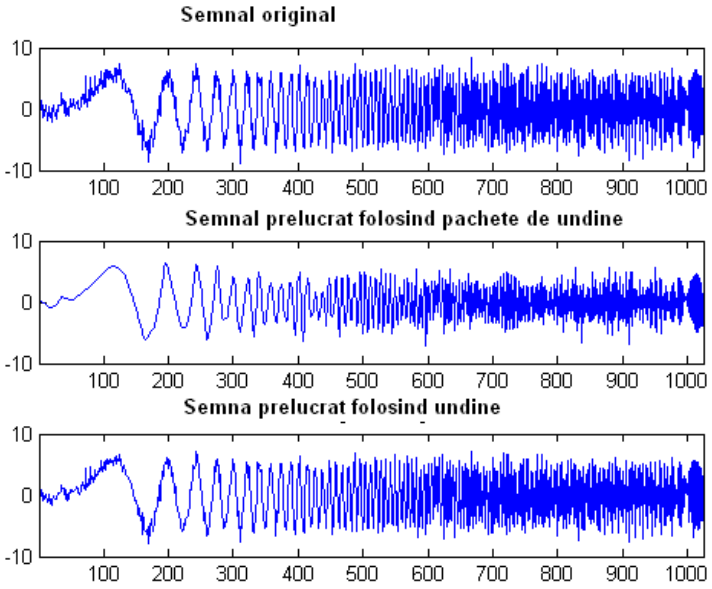

Figura 3. Rezultat cod Matlab

Concluzie

În ultimul deceniu, prelucrarea undinelor a avut o dezvoltare spectaculoasa. Initial a fost doar curiozitate matematica, acum s-a ajuns la o multitudine de algoritmi de prelucrare de semnal. Deoarece proprietăţile unui semnal se schimba in timp, este de preferat sa folosim undinele (analizarea semnalelor simultană în timp şi frecvenţă cu localizări în ferestre controlate) in prelucrarea semnalelor.

Pachetele de undine ofera mai multe informatii despre semnal decat o undina (si anume parametrii referitori la pozitie,scala si frecventa) oferind un instrument mai flexibil pe o perioada de timp pentru prelucrarea datelor. Aceste avantaje sunt păstrate, deoarece pentru fiecare pachet de undine se genereaza o biblioteca de baze. Fiecare din aceste baze ofera un mod de codificare particular a semnalelor , de memorare/modificare a energiilor si de reconstructie (cu ajutorul caracteristicilor oferite).

Această analiză timp-frecvenţă cu ajutorul undinelor/pachetelor de undine are o importanţă crucială, care deja a condus la rescrierea multor capitole ale ştiinţelor exacte, de ex. Tomografia, Recunoaşterea formelor, Electromagnetismul, Teoria vibraţiilor, Procesele stochastice, Fractalii etc. Apar posibilităţi noi – evidenţierea „zoom” – urilor, descompunerea semnalelor în „voci” (în analogie cu armonicele din cazul semnalelor periodice), noi devoalări ale fractalilor, un alt tip de „hipermicroscop” pentru structuri fizico-chimice inaccesibile direct etc.

Bibliografie:

[0]. Stanasila Octavian, Analiza matematica a semnalelor si undinelor, Ed. Matrix, bucuresti 1997
[1]. J.M.Nicolas, J.C. Delvigne and A. Lemer. Automatic identification of transient biological noises in underwater acoustics using arborescent wavelets and neural network
[2]. Yves Meyer. Wavelets and Applications. Procedings of the International Conference “Wavelets and Applications” Marseille, Mai 1989, paris, 19992, LMA/CNRS, Masson
[3]. Ingrid Daubechies. Orthormal bases of compactly supported wavelets. Communications on Pure and Applied Mathematics, XLI 909-996,1988
[4]. Ronald R. Coifam and Yves Meyer. Nouvelellts bases ortonormees de $L^2(\mathbf{R})$ ayant la strucure du systeme de Walsh, preprint, Department of Mathematics, yale University, New Haven, 1989
[5]. Mladen Victor Wickerhauser,Adapted Wavelet Analysis from Theory to Software,IEEE Press, New York, 1993
[6]. Software -Ian Kaplan, Bear Products International, www.bearcave.com, 2002.
[7]. www.mathworks.com