

# TRANZIENS JELEK VIZSGÁLATA VIRTUÁLIS MŰSZEREZÉSSEL

## INVESTIGATION OF TRANSIENT SIGNALS BY VIRTUAL INSTRUMENTATION

Dr. Tóth Lajos\*

### ABSTRACT

*This paper considers creation of a new analysis library add-on to LabWindows/CVI 7.0 that improves its analysis capabilities with transient analysis using Wavelet Filtering and Joint Time-Frequency Analysis methods. Among the available library creation methods we used the form of Dynamically Linked Libraries (DLL) that reduces the size of compiled code of final executable in contrast to the statically linked libraries. We showed the mathematical background of each analysis and the way of library creation in the development environment. We also proved the applicability of these methods with test data. This add-on can be used in analysing transient phenomenon in mechanical or electrical systems.*

### 1. BEVEZETÉS

Mérési és jelfeldolgozási feladatok esetében kézenfekvő olyan szoftverek használata mely egyszerre támogatja mindkét feladatot. A rendelkezésre álló szoftverek közül a National Instruments (NI) virtuális műszerezt megvalósító szoftverei költségkímélő megoldást biztosítanak erre a célra. A *LabWindows/CVI ANSI C* integrált fejlesztői környezete vagy a *LabVIEW* grafikus programozói környezete segítségével gyorsan fejleszhetünk mérési, tesztelési, vezérlési vagy szabályozási feladatokat ellátó virtuális műszert. A gyártó a bővítmények széles skáláját ajánlja a termékeihez, természetesen nem ingyen. A *Signal Processing Toolkit* esetében ez az ár közel megegyezik – a cikk írásának időpontjában – az alap fejlesztői környezet (*Base Development System*) árával, és több mint a teljes fejlesztői környezet (*Full Development System*) árának a fele. Ha azonban valaki ismeri jelfeldolgozási módszerek matematikai alapjait és rendelkezik némi C nyelvű programozási ismerettel,

akkor könnyen bővítheti fejlesztői környezetét ilyen vagy más jelfeldolgozási képességekkel. Ráadásul ezek az új funkciók a *LabVIEW* környezetben megvalósított virtuális műszerekben is felhasználhatók. Mi egy régi 30- napos próbaverziójú *LabWindows/CVI 7.0* fejlesztői környezetet használtunk olyan függvénykönyvtár létrehozására, mely a *Signal Processing Toolkit*-ben is megtalálható.

A tranziens jelek definíció szerint véges energiájú jelek. Jellemzésük történhet az időtartományban, a frekvencia tartományban és az idő-frekvencia tartományban. Az időtartománybeli jellemzésre a *késleltetési idő, felfutási idő, túllövés, beállási idő* illetve *lefutási idő* használatos. Frekvencia tartománybeli jellemzés a *Fourier-* vagy *Laplace-transzformációval* történik. Az idő-frekvencia tartománybeli jellemzésre az *egyesített idő-frekvencia analízis (Joint Time-frequency Analysis)* módszerek használatosak. Ezeket a módszereket a korábban említett bővítmény tartalmazza.

### 2. ANALÍZIS MÓDSZEREK MATEMATIKAI HÁTTERE

A *Fourier-transzformáció (FT)* (1) lehetővé teszi a jelek szinuszos jelkomponensek összegére való felbontását. A transzformált létezésének feltétele a jel abszolút integrálhatósága. Továbbá a jelnek csak megszámlálhatóan véges sok helyi szélsőértéke és szakadása lehet. [1]

$$\mathcal{F}\{f(t)\} = F(j\omega) = \int_{-\infty}^{\infty} f(t) \cdot e^{-j\omega t} dt \quad (1)$$

A *Wavelet-transzformáció (WT)* viszonylag új eszköz. A transzformáció segítségével a jel *Wavelet*-nek

\*adjunktus, Miskolci Egyetem Elektrotechnikai tanszék

(hullámocskáknak) nevezett bázisfüggvények összegére bonthatók fel. A waveletek  $\psi(t)$  normalizált függvények, melyeket egy „mother wavelet” (anya wavelet) függvényből eltolással és nyújtással állítunk elő. [1] A Folytonos Wavelet-transzformáció Continuous Wavelet Transform (CWT) (2) nem más, mint a jel és a wavelet konvolúciója azaz

$$CWT\{f(a, b)\} = \langle f; \psi_{a,b} \rangle = \int_{-\infty}^{+\infty} f(t) \cdot \frac{1}{\sqrt{a}} \psi\left(\frac{t-b}{a}\right) dt, a > 0, \quad (2)$$

ahol „a” a „nyújtási” vagy skálaparaméter (dilation/scale parameter), „b” az eltolási paraméter (translation parameter). [1]

## 2.1 IDŐ-FREKVENCIA ELOSZLÁSOK

Az egyesített idő-frekvencia eloszlások (Joint Time-Frequency Distributions - TFD) lehetnek lineáris vagy kvadratikusan eloszlások [2]. Az ablakozott- vagy rövid idejű Fourier-transzformáció (Short-Time Fourier Transform - STFT) és a Wavelet-transzformáció mind lineáris TFD. A kvadratikusan eloszlások a jeleket az idő és a frekvencia függvényében vizsgálják. A spektrogram (SP), azaz az STFT abszolút értékének négyzete, a scalogram (SC), azaz a WT abszolút értékének négyzete és a Wigner-Ville eloszlás (Wigner-Ville Distribution - WVD) mind kvadratikusan eloszlások.

Időben lokalizált jelek FT-ját STFT-nek hívjuk (3):

$$STFT\{f(\omega, t)\} = \int_{-\infty}^{\infty} f(t) \cdot g(t-u) \cdot e^{-j\omega t} dt, \quad (3)$$

ahol  $g(t)$  az ablakozó függvény (Window function). [2] A megfelelő ablakozó függvény kiválasztása döntő jelentőségű az analízis szempontjából mivel az idő-frekvencia felbontást ez határozza meg.

A SP (4) mindig pozitív, valós értékű idő-frekvencia eloszlás. Idő-frekvencia felbontása nem változik meg az analízis során [2].

$$SP\{f(\omega, t)\} = |STFT\{f(\omega, t)\}|^2 = \left| \int_{-\infty}^{\infty} f(t) \cdot g(t-u) \cdot e^{-j\omega t} dt \right|^2 \quad (4)$$

Az STFT és a WT a jelet az időben és a frekvenciában jól lokalizált bázisfüggvényekkel korreláltatja. Ezeknek a felbontása függ a bázisfüggvény idő és frekvencia tartománybeli kiterjedésétől.

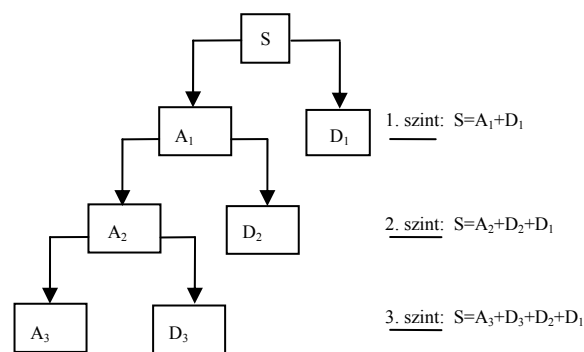
A WVD (5) a jelet önmagával korreláltatja, ezáltal jobb felbontást biztosít [2].

$$WVD\{f(u, \xi)\} = \int_{-\infty}^{\infty} f\left(u + \frac{\tau}{2}\right) \cdot f^*\left(u - \frac{\tau}{2}\right) \cdot e^{-j\xi\tau} d\tau \quad (5)$$

A WVD mindig valós mennyiség függetlenül attól, hogy a jel valós vagy komplex. Hátránya viszont, hogy az eredmény lehet negatív mennyiség is és a transzformáció nem-lineáris, azaz kovariáns az idő vagy frekvencia tartománybeli eltolásra vagy a nyújtásra. [2]

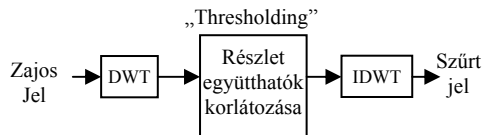
## 2.2 WAVELET THRESHOLDING

A Wavelet szűrés (Wavelet korlátozás - Wavelet Thresholding) technikája magából a WT-ből származik. A Diszkrét Wavelet Transzformáció (DWT) egy többlépcsős eljárás. [3] Egy (S) jelet átengedve megfelelően kiválasztott szűrőkön, felbontható alacsony frekvenciás közelítés (approximation) és nagy frekvenciás részlet (detail) együtthatókra ( $A_n, D_n$ ). A felbontás következő szintjén az előzőleg kapott alacsony frekvenciás komponenseket tovább bontjuk alacsony- és nagy frekvenciás jelösszetevőkre (1. ábra). A rekurzív eljárást a meg kívánt felbontási szintig lehet folytatni. A szükséges felbontási szintek számát valamilyen kritérium, leggyakrabban az ún. entrópia alapján határozzák meg. [3]



1. ábra. A jel Wavelet együtthatók szerinti felbontása

A jel Wavelet együtthatókból történő visszaállítása az előbbivel ellentétes folyamat (2. ábra). Az Inverz Diszkrét Wavelet Transzformáció (IDWT) során használt szintézis szűrőket az analízishez használt szűrőkből állítják elő. A szűrés során azokat az együtthatókat melyek egy meghatározott értéket nem érnek el korlátozzuk. Az IDWT után megkapjuk a szűrt jelet.



2. ábra. A „Wavelet Thresholding” eljárás folyamata

### 3. DINAMIKUS CSATOLÁSÚ KÖNYVTÁR (DLL) LÉTREHOZÁSA

A LabWindows/CVI lehetővé teszi a megírt függvények bináris végrehajtható állomány, statikus könyvtár vagy dinamikus csatolású könyvtár (*Dynamic Link Library – DLL*) formátumba való fordítását. [4] A fájl készülhet hibakeresési (*debug*) vagy végleges (*release*) módban. Hibakeresési módban lehetőségünk van a program lépésenkénti végrehajtására. Mi a megvalósított függvényeinket *DLL* függvénykönyvtár formátumban tároljuk. ezzel a módszerrel csökkenthetjük a futtatható állomány méretét, mivel a *DLL* nem kerül befordításba, hanem futásidőben kerül betöltésre. Másrészt mivel lefordított kódról beszélünk, az algoritmusok forráskódja részben el van rejtve a felhasználó elől. A felhasználónak definiálnia kell egy header fájlban azon függvényeket, melyek a *DLL*-ből meghívhatók. [4] Ha csak C vagy C++ forrásból szeretnénk *DLL* függvényt hívni akkor a `__cdecl` konvenciót használjuk függvényhívásra. ha más pl. Visual Basic forráskódból szeretnénk a *DLL* függvényhívást kezdeményezni, akkor a `__stdcall` függvényhívási módot kell használni. Ha CVI-on keresztül fordítjuk le a *DLL* könyvtárat, akkor a fejlesztői környezet két fájlt hoz létre. Az egyiket „*lib*” a másikat „*dll*” kiterjesztéssel. Az ún. „*import library*” a *DLL* könyvtárhoz való csatolást segíti elő. Ez tartalmazza a címtáblát (*import address table*) amelyek az egyes *DLL* függvényekre hivatkoznak. A *DLL* könyvtár projekthez rendelése történhet statikusan (implicit módon) vagy futás közben (explicit módon). Statikus hozzárendelés esetében szükségünk van az exportált függvényeket tartalmazó header fájlra, az *import* könyvtárra és a *DLL* fájlra. Futás közbeni csatolás esetében a felhasználó használhatja a *LoadLibrary*, *GetProcAddress* és *FreeLibrary* rendszerhívásokat a *DLL* függvények eléréséhez.

Az általunk megvalósított bővítmény hét analízis elvégzésére képes. Ezek a *FT*, *Cepstrális analízis*, hagyományos digitális szűrési eljárások, tranzienis jelek időtartománybeli jellemzésére szolgáló paraméterek megmérése, *STFT*, *WVD*, és a *Wavelet Thresholding*.

A header fájlunk az alábbi függvény definíciókat tartalmazza:

```

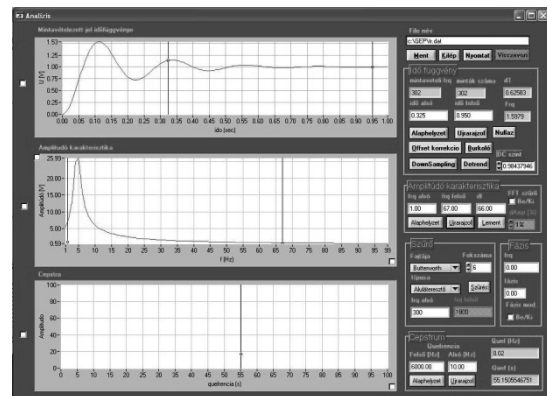
int __stdcall FFT(double *,int);
int __stdcall STFT(double *,int);
int __stdcall WT(double *,int);
int __stdcall WVD(double *,int);
  
```

Címszerinti paraméterátadást használtunk az adatok eléréséhez. A második paraméter az analizálandó adatok mennyiségét tartalmazza. Minden egyes *DLL* függvény saját grafikus felhasználói interfésszel rendelkezik, ahol a felhasználó beállíthatja az analízis paramétereit.

### 4. ANALIZÁLÓ FÜGGVÉNYEK TESZTELÉSE

A megvalósított analízáló függvények teszteléséhez elkészítettünk egy CVI programot, amely *DLL* függvényhívásokon keresztül analizálja a vizsgálandó tranzienis jeleket.

Az idő- és frekvenciatartománybeli vizsgálat felhasználói interfésze látható a 3. ábrán.



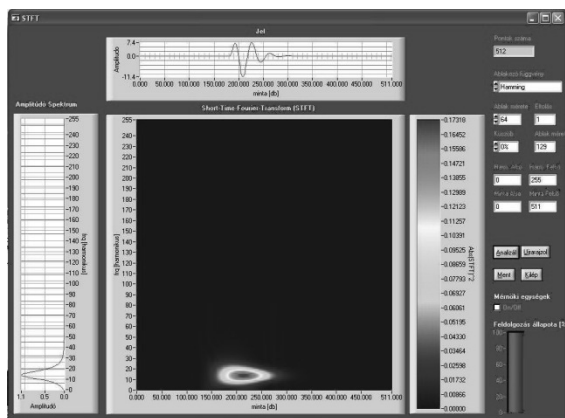
3. ábra. Idő- és frekvenciatartománybeli vizsgálat felhasználói interfésze

A felhasználói interfész lehetőséget ad a tranzienis jelek időtartománybeli jellemzésére szolgáló paraméterek meghatározására. A középső diagramon a jel frekvenciatartománybeli viselkedése látható. A legelső diagramon a jel Cepstrális analízisének eredményeit vizsgálhatjuk meg.

Az idő-frekvencia tartománybeli vizsgálatok algoritmusai közül az *STFT* és a *WVD* került megvalósításra a *DLL*-ben.

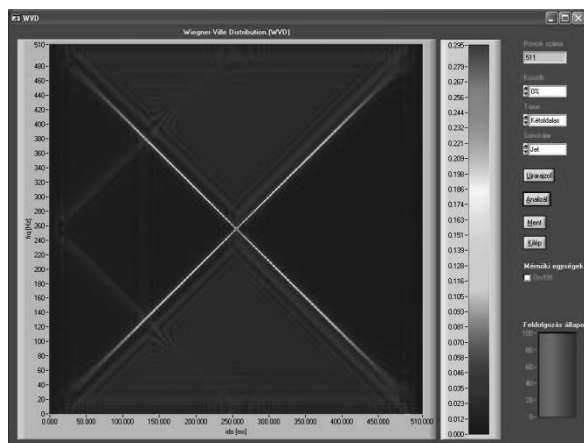
Az *STFT* felhasználói interfésze a 4. ábrán látható. A vizsgált tranzienis jel időtartománybeli viselkedése a kép felső részén, frekvencia tartománybeli alakja a kép bal

oldalán, és idő-frekvencia tartománybeli energia eloszlása a kép közepén látható. A 4. ábra tisztán mutatja, hogy a tranziens jel definíciójának megfelelően véges energiájú jel, mely jelen esetben időben és frekvenciában jól lokalizált. A felhasználó beállíthatja az ablakozó függvényt, ami lehet *Háromszög*, *Hamming*, *Hanning*, *Blackman*, *Kaiser* vagy *Blackman-Harris* típusú.



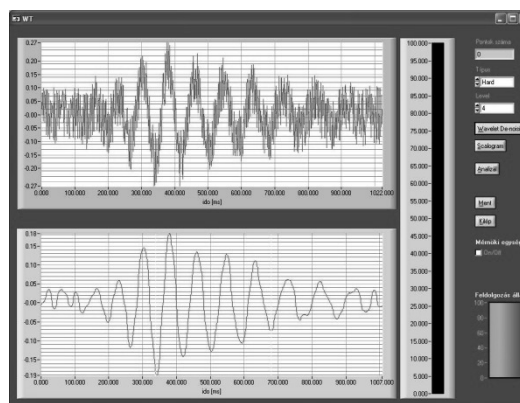
4. ábra Short Time Fourier Transformation

A WVD felhasználói interfészét az 5. ábrán mutatjuk be. Az analizált jel ebben az esetben lineáris frekvencia modulált jel 512 mintáját használtunk. A frekvenciaöket 0-tól a normalizált frekvencia feléig terjed. Az 5. ábrán a WVD eloszlással kapcsolatos érdekes jelenség figyelhető meg ami a „cross-term”-ek megjelenése. Ezek a tagok mindig előtűnnek, ha a vizsgálandó jel nem analitikus.



5. ábra Wigner-Ville eloszlás

A *Wavelet* szűrés felhasználói felülete látható a 6. ábrán. Alkalmazásával a *Wavelet Thresholding* szűrési technikát végezhetjük el a *Symmlet5* wavelet segítségével. Természetesen a későbbiekben más *Waveletek* is implementálásra kerülnek. A zajos tranziens jel az ábra felső-, a szűrt jel pedig az alsó részén látható. A felhasználó beállíthatja az a küszöb szintet és a részlet együtthatók korlátozásának módját.



6. ábra Wavelet Thresholding

## 5. ÖSSZEGRÉS

A cikkben bemutatásra került egy olyan, a LabWindows/CVI 7.0 fejlesztői környezethez készült bővítmény fejlesztésnek módja és működése, amely a tranziens jelek vizsgálatának képességével bővíti a fejlesztői rendszer analízáló képességeit. A megvalósított funkciók lehetővé teszik a tranziens jelek idő-, frekvencia- és idő-frekvencia tartománybeli vizsgálatát többek között *Fourier-transzformáció*, *ablakozott Fourier-transzformáció* és *Wigner-Ville eloszlás* alkalmazásával. A tranziens jelek szűrésére a megvalósításra került a *Wavelet Thresholding* eljárás is. Az implementált algoritmusokat Dinamikus csatolású könyvtár (DLL) formájában készítettük el.

## KÖSZÖNETNYILVÁNÍTÁS

A bemutatott kutatómunka a TAMOP-4.2.1.B-10/2/KONV-2010-0001 jelű project részeként az európai unió támogatásával, az európai szociális alap társfinanszírozásával valósul meg.

## 6. IRODALOM

- [1] S. Mallat: A Wavelet tour of signal processing, Academic Press, 1998
- [2] J. C. O'Neill: Shift Covariant Time-Frequency Distributions of Discrete Signals, PhD thesis, University of Michigan, 1997.
- [3] J.C. Goswami, A.K.Chan: Fundamentals of Wavelets, John Wiley & Sons, Inc., 1999
- [4] National Instruments: „Using Dynamic Link Libraries with LabWindows/CVI” <http://zone.ni.com/devzone/cda/tut/p/id/3341>