

Szinkronizált multi-ablakos megjelenítés jelfeldolgozó és diagnosztikai algoritmusokkal

Gárdonyi Gábor

Mechatronika, Optika és Gépészeti Informatika Tanszék
Budapesti Műszaki és Gazdaságtudományi Egyetem
Budapest, Magyarország
e-mail: gardonyi@mogi.bme.hu

Dr. Samu Krisztián

Mechatronika, Optika és Gépészeti Informatika Tanszék
Budapesti Műszaki és Gazdaságtudományi Egyetem
Budapest, Magyarország
e-mail: samuk@mogi.bme.hu

Absztrakt — Diagnosztikai célú szoftvereket az ipar minden területén alkalmaznak fejlesztés, gyártás, minőség-ellenőrzés vagy folyamatok felügyelete céljából. Ezek hátránya, hogy a fejlesztés során vagy a funkcionalitást helyezik előnybe, vagy a könnyű kezelhetőséget. Az elkészített szoftveres keretrendszer egyszerre biztosítja a felhasználó nagymértékű szabadságát, a könnyű kezelhetőséget. Belső szerkezetének köszönhetően összetett, bonyolult műveletssorozatok is automatizáltan elvégezhetők. Ezt a lehetőséget még tovább bővíti az egyedi, script alapú parancsstruktúra, mely által előre definiált, szinte bármilyen kiterjedt művelet sor leírható. A program nagy előnye a megszokott diagnosztikai rendszerekhez képest, hogy kezelőfelülete testre szabható. Szabadon jeleníthetjük meg vele egy időben tetszőleges számú, akár más-más előkészítéssel feldolgozott adatsor tetszőleges típusú eredményeit. Egy adatsor adott feldolgozását is elvégezhetjük eltérő paraméterezés mellett, miközben az összes kimenetet láthatjuk egyszerre. A szoftvercsomag számos jelkondicionáló funkciót biztosít a felhasználó számára, úgy, mint a mért adatsor vágása, szabadon konfigurálható digitális szűrő alkalmazása, értékek limitálása, újramintavételezés, stb. A beépített statisztikai számítások mellett idő- és frekvenciatartományban is lehetőség van diagnosztikai lekérdezésekre. Időtartományban csúszó ablakos algoritmusokkal lehet számíttatni, majd megjeleníteni statisztikai eloszlás-paraméterek, vagy az RMS érték időbeli változását, de szekvenciális valószínűségi hányados teszt (SPRT) is végezhető. Frekvenciatartományban többféle spektrum- és oktávanalízis számítási mód is implementálásra került, emellett frekvencia szerinti teljesítményeloszlás vizsgálatára is biztosít lehetőséget a diagnosztikai felület. További lényeges funkciók a cepstrum, STFT számítás, vagy az STFT eredmény 3D-s megjelenítése.

Kulcsszavak — rezgésdiagnosztika; analízis; jelfeldolgozás; programklón; szoftverergonomia; időtartomány; frekvenciatartomány; szögterület; valósidejű feldolgozás; szöveges parancssor; szinkronizált működés

I. BEVEZETÉS

Egy diagnosztikai rendszer végső formájában a fejlesztés szempontjából rendszerint egy kompromisszum eredménye. Egy tipikus eset, amikor egy cél érdekében specifikált szoftvert írnak, melynek kezelése egyértelmű. Ilyenkor rendszerint minimális beállítási lehetőség áll a felhasználó rendelkezésére, így a kezelő személynek sem kell mélyre ható szakmai előismeretekkel rendelkeznie. Másik esete a hasonló

célú szoftveres környezeteknek, amikor az egy általánosabb funkciót hivatott kiszolgálni, de ez már jóval több beállítási lehetőséggel rendelkezik. A kezelés sok előismeretet követel a felhasználtól. Ez utóbbi programok rendszerint hajlamosabbak instabil működésre (nem értelmezhető beállítások lehetősége), valamint kezelésük sokkal nagyobb figyelmet és időt igényel.

Az elkészített diagnosztikai rendszer felépítése, mind kezelőfelületét, mind belső struktúráját tekintve új megoldásokkal igyekszik megoldani az említett kompromisszum problémáját. Egyszerre több, válogatott adatsor, vagy teljes mérési sorozat elemzését is egy-egy kattintással meg lehet valósítani. Hatalmas előny, hogy multi-ablakos megjelenítési rendszerének köszönhetően a kezelőfelületet szabadon elrendezhetjük annak függvényében, hogy ugyanazon vizsgálatból szeretnénk egyszerre több kimenetet látni más-más beállítások mellett, esetleg más-más adatsoron elvégezve azt.

A felhasználói szabadság, miszerint bármilyen feldolgozó, lekérdező vagy megjelenítő ablak tetszőleges számú példánya előhívható, valamint, hogy eltérő metódusok kimenetei tetszőlegesen egymás mellé helyezhetők, és ezáltal azonnal összevethetővé válnak az eredmények, a belső programszerkezet bonyolultságát vonja magával. A klónozott programrészeknek, az ablakoknak, valamint az eredményként létrejött adatoknak egyaránt azonosíthatónak kell lennie. Ez szükséges a program stabilitásának megtartásához, valamint a létrejött adatok nyomon követéséhez egyaránt. Az eredmények összesítése segítségével logikai úton, vagy fuzzy alapú rendszerrel összetett döntések is levonhatók ezek alapján. [1][2]

A program számos további egyéni megoldással gazdagodott. Az adatmentést egy külön ablakból tudjuk vezérelni, ahol automatikusan történik a hivatkozott lekérdező ablakok azonosítása, állapotfelmérése majd kezelése. A multi-ablak rendszert jellemzi, hogy minden modul bármelyik másik kimenetén megjelenő új adatsort felhasználhatja, feltéve, hogy az adott adatstruktúrával kompatibilis. A diagnosztikai környezet egy egyedi, script parancsokkal vezérelhető. Ez több hasznos funkciót tesz lehetővé, melyekkel a diagnosztikai vizsgálatok ideje töredékére csökkenthető a szokásoshoz képest. Ezzel lehetővé válik a felhasználó által előre programozott jelkondicionálás, vagy

akár teljes diagnosztika elvégzése a háttérben, valamint az ablakok tetszőleges együttműködése. Ez utóbbi az implementált szinkronizációs funkció, mely képes több párhuzamos munkamenet kezelésére, és általa a munkacsoportba rendelt ablakok együttműködésének módja is testre szabható.

II. A DIAGNOSZTIKAI SZOFTVER FELÉPÍTÉSE

Egy műszaki berendezés diagnosztikája két fő lépésből áll: mérés-adatgyűjtés, valamint az adatok kiértékelése valós időben, vagy a mérési folyamatot követően. Utóbbi esetben szükség van a mintavételezett adatok tárolása, majd a fájlok megfelelő kezelésére, az adatok visszaolvasására. Mindezt az adatok teljes visszakövethetőségének megőrzése mellett kell a keretrendszernek biztosítani lehetőleg úgy, hogy a felhasználónak közben minél kevesebb dologra kelljen ügyelnie. A diagnosztika említett két fő lépésének megfelelően az elkészült szoftvercsomag is két fő modulra osztható, melynek része egy valós idejű diagnosztikára is alkalmas adatgyűjtő modul (MARIKA – Multifunkciós Adatgyűjtő Rendszer Integrált Kiértékelő Algoritmussal), valamint egy jelfeldolgozó és kiértékelő alrendszer is (PETRA – Posztprocesszáló Elemzőrendszer Teljeskörű Rezgésdiagnosztikai Analízishez). A címbe foglalt szinkronizált multi-ablakos megjelenítés a posztprocesszáló modulon keresztül kerül bemutatásra.

A. A keretrendszer felépítése

A teljes keret bemutatásához hozzá tartozna számos olyan funkció részletes ismertetése, melyek a következőkben csak megemlíthetők szintjén szerepelnek, azonban e cikk terjedelmén az ilyen szintű elemzés túlmutatna. Ezen funkciók is elengedhetetlenek egy megbízható, testre szabható, komplex keretrendszer kielégítő működéséhez. Ilyenek például a betöltő képernyő, mely teljes képernyős módban elrendezését tekintve a megjelenítő képernyőhöz igazodva mutatja a programindítás előkészületeinek státuszát. Ezt a bejelentkező felület váltja fel, vagy az opcionálisan elérhető adminisztrátori bejelentkezés. Teljeskörű felhasználó-kezelés lehetséges, mely alatt értendő új felhasználó regisztrálása, jelszómódosítás, MD5 kódolású adatvédelem, stb. A felhasználónév által lehetőség nyílik az egyéni beállítások mentésére és visszatöltésére, munkamenetek megtartására. A bejelentkezést követően a háttérben futó naplózási folyamatok is visszakövethetőbbé válnak. A felhasználónévhez rendelhető a modulokhoz tartozó hozzáférési engedélyek (pl. külön az adatgyűjtő, feldolgozó és egyéb perifériákhoz). Engedélyek egy adminisztrátori felületről adhatók, de a struktúra és az MD5 alapú egyirányú adatkódolás lehetővé teszi a licenccsere feloldást is. Az adminisztrátori felületről a felhasználók engedélyei vissza is vonhatók, regisztráció

törölhető, és a programrendszer fő könyvtárainak elérési útvonalai is ezen a felületen keresztül módosíthatók.

1) Adatgyűjtés

A teljes keretrendszer LabVIEW környezetben készült el, ennek megfelelően az adatgyűjtő modul is első sorban NI DAQ eszközökkel kompatibilis (USB-DAQ, moduláris DAQ, stb.). Ezeket automatikusan felismeri és a felhasználó által definiált mérési beállításoknak megfelelően konfigurálja. A keretrendszer belső struktúráját tekintve fel van készítve más kommunikációs protokollt igénylő eszközök kezelésére is (FPGA, PXI, RS-232/485 kommunikációs A/D átalakítók, stb.) A mérés közbeni adatvesztés elkerülése érdekében – melyet a számítógép esetleges túlterheltsége okozhat – az adatgyűjtő és a posztprocesszáló modulok párhuzamos futtatása nem lehetséges. Ennek ellenére a közvetlen átjárás biztosított a két felület között. Mindkét modul alapértelmezett módon tdms multi-fájlokat használ, mely jól strukturált, bináris kódolásának és a hozzá tartozó indexfájlnak köszönhetően a gyors adatkezelés lehetséges. A multi-fájl rendszer azt biztosítja, hogy kezelhető méretű adatsomagokat kapjunk az adatrögzítést követően. A fájl tördelése a felhasználó által meghatározott maximális fájl méret alapján történik. Elsődleges célja ennek a megoldásnak, hogy elkerüljük hosszú mérések és nagy mintavételei frekvencia alkalmazása esetén a több GB méretű nehezen kezelhető fájlokat. Szélsőséges esetben akár e-mail-kompatibilis méretben is tárolhatjuk a nyers adatsorokat.

2) Posztprocesszálás

A jelfeldolgozó modul úgy került kialakításra, hogy azzal mérési adatokat a lehető legegyszerűbben és leggyorsabban ki lehessen értékelni. A különböző funkciók a menürendszerből (1. ábra) indíthatók el külön ablakban. A menü az ergonómia szem előtt tartásával lett elrendezve, kategorizált, gombnyomásra elérhető felületeken keresztül tallózhatunk az alkalmazható eljárások között. A menürendszer általában igaz, hogy a többi működő funkció, megnyitott ablak státuszáról a felhasználó felé a gombok állapotán, vagy esetenként az állapotsoron keresztül szöveges üzenet formájában visszajelzést biztosít. A kategorizált funkciók motorja minden esetben a később részletesebben tárgyalt multi-ablak rendszer, mely lehetővé teszi, hogy bármely funkciók futhatnak egymással párhuzamosan, valamint, a forráskódok klónozása által több azonosítószámmal ellátott példány is kezelhető egyszerre ugyanazon funkciónak. Vannak olyan kezelőablakok, melyek a tallózott kategóriától függetlenek és csak egyetlen példány futhat belőlük. Ilyen a fájlbetöltés, mentés, „Sync” (megnyitott ablakok szinkronizációjának kezelőablaka) és a lejátszás. Ezek a kezelőablakok ebben a fejezetben kerülnek rövid taglalásra, részletesebben csak a multi-ablak rendszerű funkciókról lehet majd a következő fejezetekben olvasni. A kezelőfelület



1. ábra: Posztprocesszáló program menüje

további gombjai a programmal kapcsolatos műveletekre vonatkoznak (pl. összes nyitott ablak adatainak törlése, ideiglenes adatok ürítése, ugrás a mérőprogramra, vagy kilépés).

a) *Betöltés* – A mérési adatok kiválasztására egy olyan kezelőfelület szolgál, mely automatikusan kilistázza a mérési mappákat, a benne található fájlokat, valamint az abban tárolt adatsomagokat. Lehetőség nyílik egy adott fájl, multi-fájl vagy több fájl beolvasására is. A vizsgálni kívánt fájl kiterjesztése (a mérési mappában található fájlok listáját a program ez alapján szűri meg), valamint a lefűzni kívánt csatornák is ezen az ablakon belül jelölhetők ki. A betöltési folyamatról itt is a kezelőgomb állapotával kap a felhasználó visszajelzést, valamint a betöltött adatsorok visszaigazolása a kiolvasott csatornák neve által történik. A felületen megjelenik a szinkronizációs funkció egy lehetősége is: egy „SyncGroup” címkét viselő listába megadhatunk számokat, melyekkel az adott számú szinkronizációs csoportra utalunk (ld. később). Amennyiben a lista nem üres, fájlbetöltés esetén a megjelölt csoportba tartozó összes háttérben futó kezelőablak kap egy broadcast parancsot, miszerint azok a betöltött fájlak és szinkronizációs beállításainak megfelelően frissítik tartalmukat. A szoftver több fájlformátum kezelését támogatja, valamint fel van készítve nagyméretű fájlok kezelésére is. A program egy egyedi tdms alapú „temporary” fájlkezelést alkalmaz a belső adatkezeléshez. Ezzel értem el, hogy folytonos adatkezelő algoritmusok alkalmazásával – merevlemez méretétől függően – gyakorlatilag bármilyen méretű adathalmazt képes betölteni, tdms fájlra konvertálni, majd később feldolgozni. Az ideiglenes fájl alkalmazása minimalizálja a memória allokációk méretét, amivel a szoftverműködési sebessége is megnövekedett.

b) *Mentés* – Diagnosztikai vizsgálatokat követően a kapott eredményeket rendszerint archiválni kell. Ez esetben a megjelenített grafikonok képei, a feldolgozott vagy módosított adatsorok, valamint a lekérdezések numerikus, esetleg logikai típusú kimeneteinek együttes, vagy szelektív tárolási módszerei a mentés ablakon keresztül lehetségesek. Bármely aktív lekérdező/feldolgozó ablak kiválasztható, de egy szabadon összeválogatott ablakcsoport eredményei is egyszerűen fájlba menthetők. A kezelni kívánt felhasználói ablakok egy listában választhatók ki az ablak neve alapján. Az azonosításra szolgáló név a multi-ablakos rendszer miatt még egyedi sorszámmal is el van látva, így azonos funkcionalitású ablakok párhuzamos futtatásakor is lehetőség van a szelektív mentésre. A program csak az aktív állapotú, adatot tartalmazó ablakok nevét ajánlja fel, így elkerülhető az üres fájlok, adatsort nem tartalmazó megjelenítők képeinek tárolása. A kijelölt elemek által tartalmazott adatsorok, illetve képek mentése külön-külön és együttesen is indítható. Az adattárolást ebben az esetben is jellemzi, hogy több adatformátumot támogat, érdemes a további feldolgozási célok figyelembe vételével megválasztani a szükséges kiterjesztést. A program képek esetén bmp, emf, eps, jpeg, png, numerikus adatsoroknál pedig bin, csv, tdms, tsv, txt, wav, wavtxt, xls és xml formátumokat ajánl fel. A fájlok mentése automatikusan egy strukturált

mappaszerkezetben történik, melyet a diagnosztikai rendszer magától létrehoz. A fájlnevek generálásakor a teljes visszakövethetőség megvalósul, tartalmazzák a mérés azonosítóit, az adatsorokon elvégzett műveleteket és azok releváns paramétereit, valamint amennyiben már az adott fájl létezik, az felülírás elkerülése érdekében kiegészítő sorszámokat is rendel a mentési algoritmus a fájlnevek végére.

c) *Szinkronizáció* – Ebben az ablakban nyílik lehetőség az aktív lekérdező ablakok szinkronizációs csoportokba (továbbiakban SyncGroup vagy SG) való rendezésére. A kezelőfelület a következő funkciókat biztosítja:

- Kiválasztott SG tartalmának megtekintése (egy adott csoporton belül összeszinkronizált ablakok)
- Új SG létrehozása
- Kiválasztott SG törlése, összes ablak felszabadítása
- Ablak hozzáadása SG-hez (csak más csoporttól független ablakokat lehet kiválasztani egy automatikusan frissülő legördülő menüből)
- Kiválasztott ablak felszabadítása
- Ablak áthelyezése másik SG-be
- Ablak szinkronizációs módjának módosítása
- Lista egy elemének felülírása
- Maximum
- Terjedelem
- Elemek száma
- Hossz

A program rugalmasságát prezentálja, hogy bármennyi SG létrehozható, így a párhuzamosan futó, külön szinkronizált csoportok mennyisége nem ütközik szoftveres korlátokba. Minden ablakhoz rendelhetünk egy szinkronizációs módot (továbbiakban SyncMode vagy SM). A szinkronizációs módok részletezése későbbi fejezetben olvasható.

d) *Lejátszás* – A rezgésdiagnosztikában elterjedt gyakorlat a mért gyorsulásjelek audio jellé való konvertálása is. Ennek megfelelően a programcsomag tartalmaz egy egyedi fejlesztésű médialejátszó modult. Ez a többihez hasonlóan képes szinkronizált működésre, így bármely algoritmus eredménye egy kattintással elérhető és meghallgatható. A lejátszó szoftver a mérési adatsort a hangkárttyával kompatibilis mintavételezési frekvenciával újra-mintavételezi, majd ezzel az adatsorral tölti fel folytonosan a lejátszó puffert. A lejátszó a következő funkciókat tartalmazza: betöltött adatsor kurzorokkal kijelölhető jelrészletének lejátszása; folytonos lejátszási funkció; külön ablakban paramétereizhető, egyedi fejlesztésű Equalizer (hangszínszabályzó – továbbiakban EQ). Az EQ választható típusú szűrőt használ (Elliptic, Butterworth, Chebishev, Inv. Chebishev). A sávszűrők oktáv szélessége a 4 oktávostól az 1/6 oktávorig 6 fokozatból választható ki. A sáv szélességek,

valamint a sávszűrő vágási meredeksége paraméterként adható meg. A sávszűrők frekvencia-átviteli karakterisztikája az összesített átviteli karakterisztikával együtt grafikonon ábrázolásra kerül. Az esetleges torzult hangzást okozhatják számítási hibák, vagy hardveres korlát is. A hangkártya esetleges limitálása, valamint az EQ esetleges paraméterezéséből adódó számítási hibák a felhasználó felé kijelzésre kerülnek.

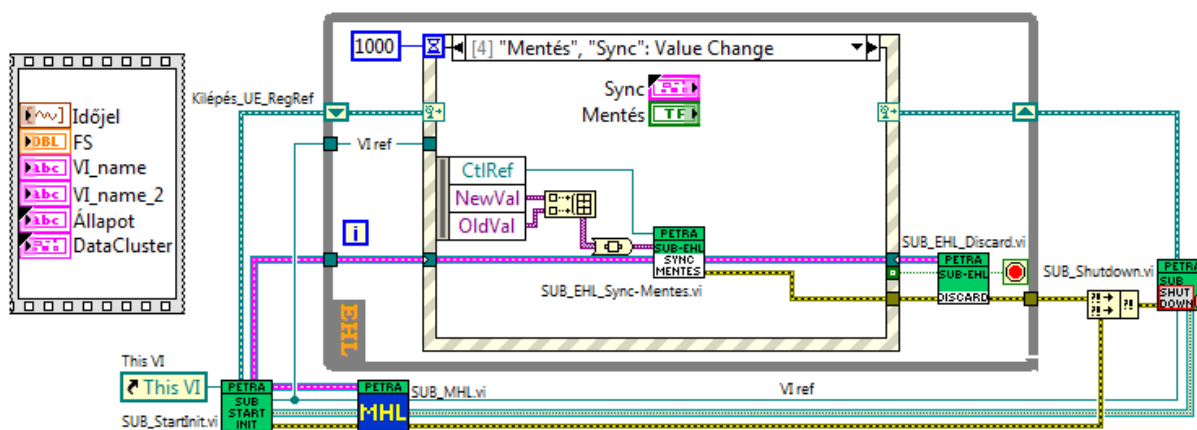
B. Lekérdező ablakok egységes struktúrája

A lekérdező ablakok gyors fejlesztetősége érdekében egy olyan rugalmas rendszer került implementálásra, melyben erősen elkülönülnek az általános ablak- és adatkezelési funkciók, valamint a lekérdező ablakhoz tartozó egyedi számítási és ablakkezelési blokkok. A struktúrát tekintve EHL/MHL háttérü alprogramokról beszélhetünk (2. ábra) (EHL – Event Handler Loop – Eseménykezelő ciklus; MHL – Message Handler Loop – Üzenetkezelő ciklus). A kezelőfelületi tevékenységek leereagálása rendszerint az EHL ciklussal történik, melyben szintén megtalálhatók a minden funkció esetén fellelhető egységesített funkciók (timeout; ablak üritése; kiválasztott jelcsoport módosítása; mentés; sync; ablak bezárása, kilépés, EHL leállítása). Az egységesített funkciók a programduplikátumok elkerülése érdekében subrutinokként épülnek be. Az ilyen módú egységesítés által elérhető, hogy a keretrendszerbe integrált számos lekérdező algoritmus esetén is, amennyiben módosítás szükséges a fejlesztés során, elegendő csak egy esetben az adott funkció főpéldányát átírni az összes ablak-kezelő rutin helyett. Természetesen a hagyományos programozási módok általában nem teszik lehetővé az ilyen szintű rugalmasságot. A LabVIEW környezetben lehetőség van a kezelőfelületek által tartalmazott vezérlő vagy kijelző elemekre referencia alapján hivatkozni. A referencia alapú programozás által lehetőség nyílik az elemek szubrutinból, vagy azonos prioritású, párhuzamosan futó programból való vezérlésre. Ez a programozási stílus a hagyományos módoknál sokkal nagyobb odafigyelést, hozzáértést igényel, de ezáltal nagy

referenciáit tartalmazza. Fontos, hogy a referenciákat a programindításkor frissíteni kell, mert azok az adott program leállításakor elévülnek. A referenciaazonosítás és hasonló egyszerűbb programrészek minden ablak esetén szükségesek. Az ablakok gyakran eltérő tartalmú programrészeket igényelnek, más adattípusok kezelése, eltérő kezelőfelületi elemek miatt. Az ilyen szekciók is át lettek ültetve egy egységesített struktúrába, ami alapján például a program magától felismeri, hogy a DataCluster mely eleméhez melyik objektum referenciáját kell rendelni. A fejlesztés során ezeket a részeket már érintetlenül is lehet hagyni, csak egy DataClustert kell definiálni, amit onnantól kezdve a program már önállóan felismer és kezel.

Az EHL mellett minden program tartalmaz egy további inicializáló programot, mely egyéb általános indításkor elvégzendő feladatot lát el. Ilyenek az EHL és MHL ciklusok szükséges bemeneteinek előállítás, de ez a programrész ellenőrzi a hívó program funkcióját, állapotát, nevét, valamint azt is, hogy klónozott program hívta-e meg. Ezeknek megfelelően kitölti az ablak azonosítására szolgáló címsorokat majd előtérbe hozza az ablakot. Az inicializáló programon kívül minden feldolgozó programban fellelhető egy folyamatot lezáró rutin is, mely az előbb részletezettekhez hasonló általános feladatokat lát el. A programfutást egy szubrutinba foglalt általános állapotgép biztosítja, mely az EHL-lel paralel módon fut a háttérben. Az ún. MessageQueue vezérelt állapotgép működését, lehetséges a hozzá tartozó programból, párhuzamos ciklusból, de külső programokból is befolyásolni. Az ehhez szükséges publikus információkat természetesen globális változóknak érdemes tárolni. Mivel az ilyen típusú általános információk a programindítást követően a fentebb említett inicializáló rutinok által megosztásra kerültek, ettől kezdve bármely program a hierarchikus viszonyoktól függetlenül hozzá tud férni és beavatkozni egy tetszőlegesen választható másik programrész működésébe.

C. A script alapú funkciók



2. ábra: Egységesített EHL/MHL struktúra

modularitású, átlátható, sokrétegű programstruktúrát hozhatunk létre. Minden ablakhoz tartozik egy ún. "DataCluster", mely a felület meghatározott elemeinek

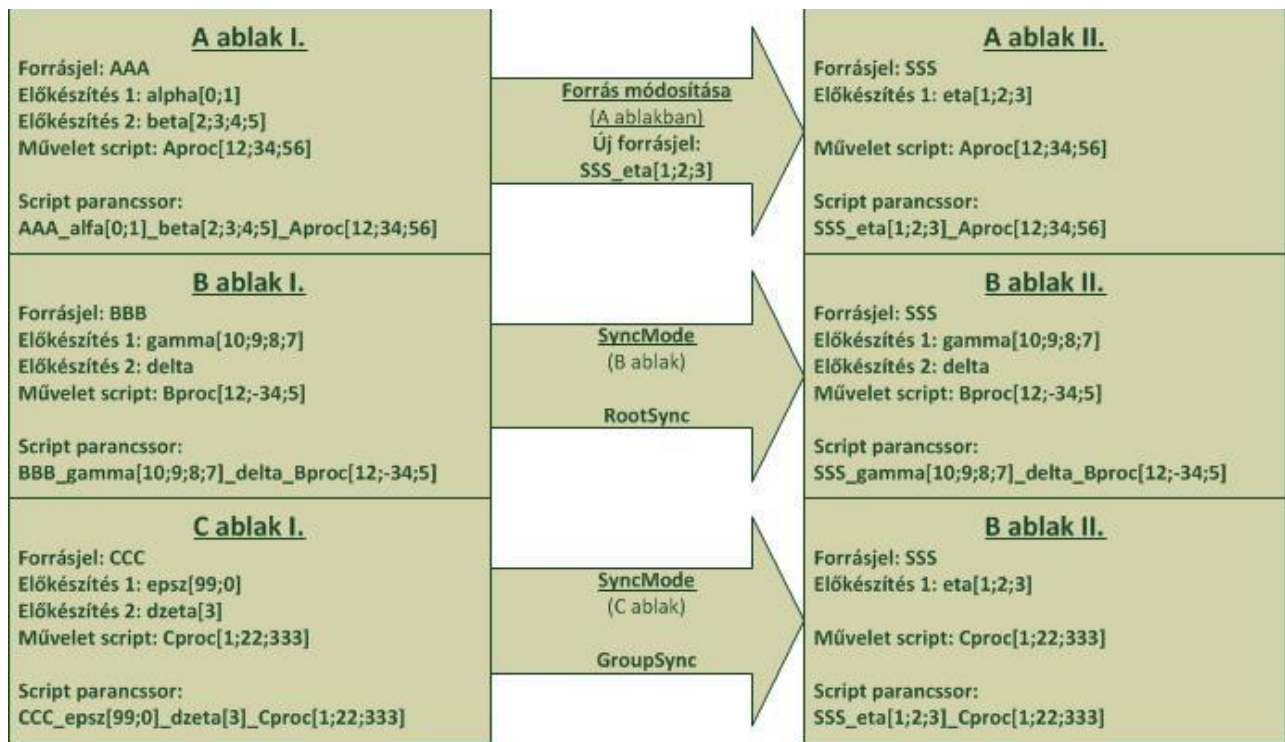
Már említésre került, hogy minden felhasználói program tartalmaz egy egységes MHL elemet. Ez a ciklus egy állapotgépet tartalmaz, melyet az összes leprogramozott

lekérdező ablak használ. Az állapotgép ún. MQ vezérelt, amely habár úgy lett kialakítva, hogy bármilyen adattípust képes egy paranccsal együtt az MHL felé közvetíteni, mégis célszerű egy adott parancs esetén egy hozzá tartozó adattípust definiálni. A „Frissítés” parancs mellé a struktúrába egy szöveges parancssor érkezik, mely definiálja a vizsgálandó jelet, a rajta elvégzett feldolgozási eljárások típusát, paramétereit és a műveletek sorrendiségét, illetve az ablak saját script parancsát. Tehát 3 fő részre lehet osztani az így összeállt script sort:

- Forrásfájl azonosító
- Elő-feldolgozási műveletek
- Lekérdező ablak beállításából előállított script

korlátozása, valamint egy adott parancsot is kiadhatunk akár kevesebb paraméterrel. Amennyiben a vártnál kevesebb paraméter érkezik be, a program 0 értékekkel helyettesíti a sorban utolsó tagokat, ha több van, akkor pedig nem veszi figyelembe őket.

A lekérdezés kimeneti azonosítójának generálásakor a módszer a forrásjel nevét veszi alapul (jelazonosító + a rajta elvégzett megelőző műveletek), majd egy alulvonást követően a végére fűzi saját script kódját. Az így kapott karaktersor lesz a számított adatsor azonosítója. Az új adatsor, amennyiben az ablak egy SG tagja, tárolásra kerül a merevlemezen egy ideiglenes fájlban. Ezáltal megosztásra kerül az összes többi ablakkal és azok – amennyiben a kimeneti adattípus kompatibilis – tovább tudnak dolgozni vele.



3. ábra: RootSync és GroupSync szinkronizációs beállítások logikája

Minden ablak által végzett művelethez tartozik egy előre megadott script kód, valamint az eljárás pontos definiálásához tartozó néhány egész típusú numerikus paraméter. Amennyiben logikai, vagy egyéb típusú paraméterrel lehetne a funkciót pontosan leírni, ez esetben is numerikus alakra kell hozni, illetve lehetőség van pl. törtrészek megadására is, de ez esetben előtte egy konstanssal való szorzással nagyságrendet növel a háttér algoritmus és ezt követően kerekít egészre. Egy művelet leírása a rendszeren belül a következő szintaktikával valósul meg:

```
proc[param1;param2;param3]
```

Ebben az esetben a „proc” a műveletre vonatkozó szöveges parancs, a „param1”, „param2”, „param3” helyére pedig numerikus egész típusú értékek kerülnek. A rendszer a parancssorokat rugalmasan kezeli, az alkalmazott paraméterek száma lehet különböző, a paraméterek számának nincs felső

D. Lekérdező ablakok szinkronizálható működése

Az SG-kezelő felület rövid bemutatásakor már említésre került, hogy több SM is elérhető minden lekérdező ablak esetén. Ez a két mód az ún. RootSync (továbbiakban RS) és a GroupSync (továbbiakban GS). A szinkronizációs módok az ablakok együttes viselkedését befolyásolják, pontosabban az együttműködés lehetőségeit bővítik.

A szinkronizációs működés lényege, hogy az ugyanazon SG tagjai közül, hogy ha az egyikén egy frissítést magával vonó esemény történik (pl. forrás vagy paraméter módosítása, kurzor áthelyezése egy grafikonon, stb.), az SG többi tagja is ennek az eseménynek megfelelően frissíti tartalmát (3. ábra). A szinkronfrissítést kiváltó ablak szempontjából nem lényeges, hogy az melyik SM-ban van, ez a SG passzív ablakainak működésére van csak hatással. Amennyiben egy adott ablak RS módban van és kényszerített frissítés parancsot kap, csak a nyers mérési adatsorra való hivatkozást módosítja, az utó-feldolgozó eljárások, azok paraméterei, valamint a saját beállítások is maradnak a régié. Ez a mód abban az esetben alkalmazandó, ha több mért adatsoron szeretnénk gyorsan, hatékonyan megtekinteni ugyan azon lekérdezések eredményeit. AGS mód több adatot örökít a passzív ablak számára, ebben az egész elő-feldolgozott jelet örökli, és csupán a saját paramétereit tartja meg. Ez a módszer már okozhat egyes esetekben kompatibilitási problémákat, de működése a RS módhoz képest gyorsabb, hiszen ekkor már rendelkezésre áll az előkészített adatsor is. Az RS módban való vizsgálatok esetén általában szükség van az előkészített forrás háttérben való automatikus számítására is.

E. Automatikus jelfeldolgozás

Az előző fejezetben, a szinkronizációs módok részletezésekor került fel, hogy egyes RS lekérdezések esetén nem mindig áll rendelkezésre a feldolgozni kívánt, kondicionált adatsor. Ilyen esetekben szükség van rá, hogy a szoftver ezeket saját magának le tudja generálni. A hiány felderítésének érdekében a frissítéshez szükséges számítások elvégzése előtt egy ellenőrző algoritmussal vizsgálja felül a parancsot a program. A script parancsot lépésről lépésre bővíti a nyers adatfájl azonosítójával kezdve, és a merevlemezen ideiglenesen tárolt adatok között keresi azt. Amennyiben az nem található, mert nem került még sor annak kiszámítására, vagy a számításakor még nem volt az adott ablak egyik SG tagja sem, a döntési logika kiküldi a parancsot az MHL állapotgépnél. Itt jelenik meg az egységesített MHL ciklus újabb nagy előnye, hogy mivel minden metódus ugyan azt a programot használja, mindegyik számára elérhetővé válik az összes számítási algoritmus, mely a komplett diagnosztikai rendszerben implementálásra kerül. A hiányzó adatsor utolsó script tagja alapján képes azonosítani, hogy mely művelet eredménye hiányzik, a szükséges paraméterek azonosítása magától értetődő. Az újonnan kapott adatsor tárolásra, valamint megosztásra kerül, majd az ellenőrző algoritmus folytatja a hiányzó, de szükséges adatsorok keresését, majd pótlását.

A vázolt rendszer már magába foglalja, hogy a szoftver képes hiányzó adatsorokat magától detektálni, majd pótolni.

Képes a háttérben megjelenítő felület nélkül számításokat végezni, majd az eredményeket a merevlemezen fájlba menteni, ezáltal megosztani azokat. Ezek a tulajdonságok már elegendőek is ahhoz, hogy egy megjelenítés nélküli diagnosztikai eljárást csak script alapon futtassunk, és a részletes elemzés helyett csak a komplett diagnosztikai eljárásorozat végeredményét kérdezzük le. Ezek a jellemzők eredményezik azt is, hogy nagyon egyszerűen lekérdezhető akár kiterjedt mérési sorozatokra is ugyanazon lekérdező eredményesora.

III. IMPLEMENTÁLT FELDOLGOZÓ ALGORITMUSOK

A következőkben felsorolt algoritmusok mind külön-külön ablakban megnyíló, központilag szinkronizálható működésű alkalmazások. Ezek mindegyikének adat- és képmentési funkciói a fentebb említett minden fájlformátumban megoldott, illetve mindegyik a közös elérési ideiglenes fájl használja adatmegosztás céljából. Közös bennük továbbá, hogy kezelőfelületükön van egy kevésbé kiterjedt, egyéni SG kezelő vezérlőelem, mellyel csatlakozni lehet valamely SG-hoz, de a tagság meg is szüntethető. Erről a felületről a SM nem módosítható, alapértelmezettként RS módban kerül regisztrálásra. A kezelőelem a SG-kezelő program hívógombjának állapotával, valamint a SG azonosítószámának megjelenítésével is jelzi, hogy az adott lekérdező ablak rendelkezik-e tagsággal.

A felsorolt ablakok kialakításakor fokozott szempontot képviselt a szoftver ergonomikus kialakítása, így gyakran fordulnak elő redundáns paraméterezési lehetőségek. Például a felhasználó egy adott paramétert grafikus módon, vagy több formában numerikusan is megadhat. Így a számára legkényelmesebb beviteli módot választhatja ki. A beviteli elemek működése minden esetben szinkronizálva van, egyik módosítása esetén a többi kontrolelem tartalma is változik.

Megjelenítő ablakok

- Időjel – Egyszerű megjelenítő ablak a választott adatsorok megjelenítésére. A betöltött adatsor összes csatornáját megjeleníti egyszerre az idő függvényében. (5. ábra)
- Időjel multivizet – Megjelenítő ablak, mely a megosztott jelsorokat csatornánként ajánlja fel kiválasztásra. Jelcsoportokat válogathatunk össze tetszőleges számú grafikonra, melyek közösen megjelenítésre kerülnek. A grafikonok skálája mindig az utoljára aktivizált (pl. kattintással) megjelenítő által tartalmazzott adatsorhoz igazodik, így akár különböző skálázást igénylő jelek is releváns módon összevethetők.
- Spektrum – Egyszerű megjelenítő ablak a választott adatsorok megjelenítésére. A betöltött adatsor összes csatornáját megjeleníti egyszerre a frekvencia függvényében. Az időjel megjelenítő ablaktól való jelentős eltérés, hogy szabadon konfigurálható a frekvencia és az amplitúdó tengely is. Szabadon választható, hogy logaritmus vagy lineáris skálázáson szeretnénk-e az adatok kirajzolását.

- Spektrum multinézet – Az időjel multinézethez hasonló funkcionalitású ablak, csak spektrumok adatsorainak együttes megjelenítésére.

A. *Jeletőkészítő műveletek*

- Vágás – Egy jel vizsgálatakor ez a modul egy részletet emel ki az adatsorból, majd ezt további feldolgozásra bocsátja. A vágási paraméterek grafikonon kurzorral, vagy 4 módon numerikus értéként is megadhatók
 - Kezdőpont [s] – Hossz [s]
 - Kezdőpont [db] – Hossz [db]
 - Kezdőpont [s] – Végpont [s]
 - Kezdőpont [db] – Végpont [db]

A vágási algoritmus fejlesztése során fokozott figyelem fordult a kis memóriahasználatra, így több GB méretű fájlokat is gond nélkül és gyorsan kezel a modul.

- Szűrés – Egy jel, vagy jelrészlet digitális szűrését teszi lehetővé. választható Butterworth, Chebishev, Inv. Chebishev, Elliptikus, valamint Medián szűrő. Előbbi négy esetén elérhetők alul-, felüláteresztő, sávszűrő és sáváteresztő változatok is. A szűrőt definiáló főbb paramétereket grafikonon és numerikus elemeken is meg lehet adni. A program számítja és megjeleníti a digitális szűrő amplitúdó-menetét, fázis-menetét, valamint a szűrt jelet. (7. ábra)
- Limit – Alsó és felső küszöbszintek állíthatók az időjelre, melyek ez esetben is grafikonon kurzorral, valamint numerikusan egyaránt megadhatók. Megjelenítésre kerül az időjel a küszöbszint határokkal (alsó és felső határok címkékkel ellátva). Megjelenik továbbá az adatsor minimuma, maximuma, átlagértéke. Választható funkciók a szimmetrikus határdefiniálás és az offset kompenzáció.
- Újramintavételezés – Az adatritkítástól eltérő eljárás, mely segítségével az adatsor információtartalma jelentősen nem változik, viszont a tartalmazott adatok száma igen. A kívánt új mintavételezési frekvencia és a pontillesztés módjának definiálását (pl. interpoláció, görbeillesztés) követően számítja egy algoritmus az új jelsort.
- Ritkítás – A vizsgálat szempontjából felül-

mintavételezett adatsorok esetén hasznos modul. Az adatritkítás mértéke több beviteli módon, szabadon választható.

- AR szűrés – Egy autoregressziós modell paraméterezhető felépítését követően a vizsgált jel „kifehéríthető”. A determinisztikus komponenseket eliminálja, ezzel együtt az adatsort nullaközepűvé is teszi.

B. *Statisztikai feldolgozás*

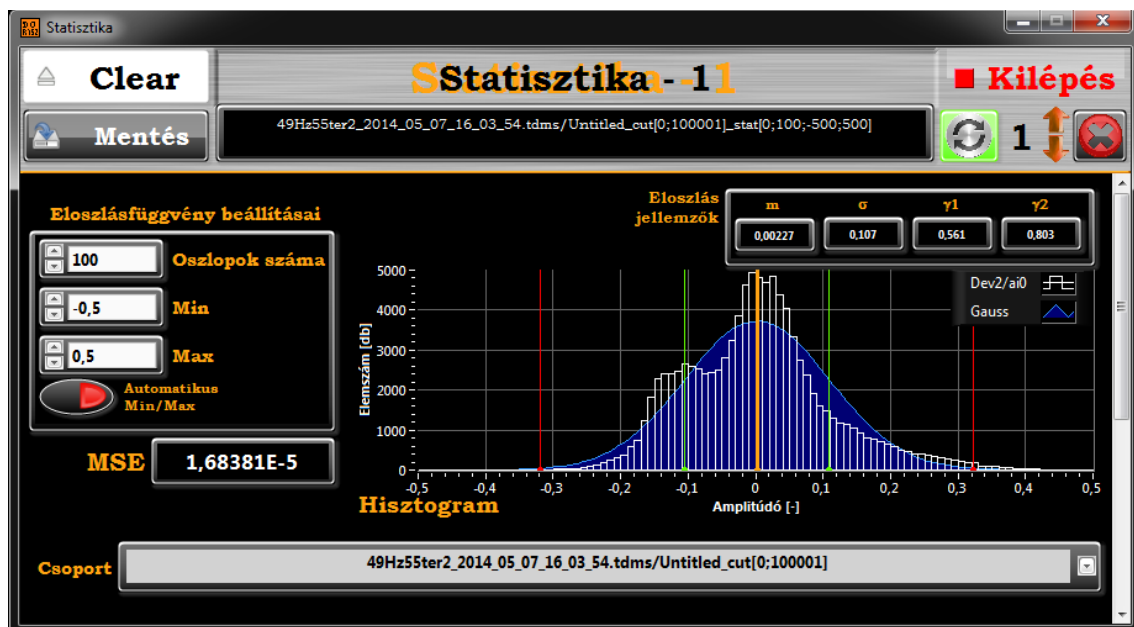
Ebben a modulban több felületen lapozhat a felhasználó, melyekből az elsőt megjelenítésre kerül a jel paraméterezhető hisztogramja (állítható az oszlopok száma, min, max; vagy választható automatikus paraméterezés is). A hisztogram háttérben megjelenik a vizsgált jellel megegyező szórású és középértékű Gauss eloszlásfüggvény, valamint szemléltetésre kerülnek a várható érték, a szórás és 3xszórás intervallumok is (4. ábra). A hisztogram mellett megjelennek a ferdeség és lapultság értékek, valamint a Gauss eloszlástól való átlagos négyzetes hiba (MSE) nagysága.

A második lapon a következő statisztikai adatok jelennek meg:

- | | |
|--------------------|------------------------------|
| • Átlag | • Tapasztalati szórás |
| • Medián | • 3x Tapasztalati szórás |
| • Harmonikus közép | • Variancia |
| • Ferdeség | • Harmadik momentum |
| • Lapultság | • Negyedik momentum |
| • RMS | • 0.1-es kvantilis |
| • RMS ² | • 0.5-ös kvantilis |
| • Minimum | • 0.9-es kvantilis |
| • Maximum | • Kezdeti időpillanat |
| • Terjedelem | • Mintavételezési frekvencia |
| • Elemek száma | • dt |
| • Hossz | |

C. *Időtartománybeli műveletek*

- Csúskeresés – Szemléletes megjelenítése a definiált



4. ábra: Statisztikai feldolgozó ablak megjelenítő felülete

küszöbszint felett talált csúcsoknak. Az időjelben kirívó jelzéssel jelennek meg a beállított határok, valamint az interpolációs algoritmussal számított csúcsok/völgyek helyei. Kijelzésre kerül a detektált csúcsok száma, a jel maximum és minimum értékei. Választható funkciók a szimmetrikus határdefiniálás és az offset kompenzáció. (6. ábra)

- Eloszlás (t) – Csúszó ablakos eljárással implementált algoritmus, mely a jel statisztikai paramétereinek időbeni változását számítja és jeleníti meg. A jel első négy momentumából számított átlag, szórás, ferdeség (skewness) és lapultság (kurtosis) értékeit számítja ki a kiemelt jelrészletre. Az ablak mérete és a csúsztatás mértéke megadható 3 formátumban (s, μ s, dt), a számított/megjelenített adatsorok szabadon kiválaszthatók egy felajánlott listából a számítógép tehermentesítése érdekében. Az eljáráshoz folytonos adatkezelésű algoritmust alkalmaz a szoftver, mely lehetővé teszi kiterjedt adatsorok gyors és memóriatakarékos feldolgozását. A számítási folyamat bármikor abortálható. (9. ábra) [3]
- RMS(t) – Csúszó ablakos eljárással implementált modul. A jel teljesítménytartalmának időbeni változását számítja és jeleníti meg. Az ablak mérete és a csúsztatás mértéke megadható 3 formátumban (s, μ s, dt). Megjelennek a felületen az RMS(t) függvény átlagértéke, maximuma és a maximum időben vett helye. Az Eloszlás(t) algoritmusnál leírtakhoz hasonlóan kiterjedt méretű adatsorok feldolgozása is lehetséges. (8. ábra) [4]
- SPRT(t) – Csúszó ablakos eljárással implementált modul, mely a jel paramétereizhető szekvenciális valószínűségi hányados tesztjéből (SPRT) kinyert „Lambda” függvényt számítja és jeleníti meg. A program felületén megjelennek a döntési algoritmus által detektált riasztások száma, a Lambda függvény maximuma és a maximum időben vett helye. Az algoritmus optimalizációjának eredményeképpen ez szintén memóriatakarékos eljárás, kiterjedt adatsorok feldolgozása esetén sem ütközik problémába. (10. ábra) [5]

D. Frekvenciartománybeli műveletek

- Spektrum – Választható módon amplitúdó-, teljesítmény- és teljesítménysűrűség spektrumok is lekérdezhetők. A számítás módját tekintve lehet egy lépéses vagy átlagolós FFT algoritmust választani. Utóbbi egy egyedi fejlesztésű adatvesztés-minimalizált átlagolási eljárás, melyhez a minimális átlagolások számát lehet paraméterként megadni. A program szerkezetének kialakítása lehetővé teszi, hogy az számítás közben a részeredményeket a kezelőfelületen folyamatosan frissíti. Az ablakban megjelennek továbbá a vizsgálat alá vetett időjel főbb paraméterei (dt, elemszám, terjedelem), a spektrumszámítás eredményének főbb paraméterei (df, elemszám, maximális amplitúdó

értéke, valamint annak helye a frekvenciatengely mentén), a számított spektrum adatsorának summája és frekvencia szerinti integrált értéke. A megjelenítő tengelyei testre szabhatók (lin/log skála, tengelyen feltüntetett numerikus értékek tizedesponosság, automatikus/manuális skálaigazítás, láthatósági beállítások, stb.)

- Oktáv analízis – Előnye, hogy a jel teljesítménytartalmának frekvencia szerinti tömörüléseit a spektrumnál sokkal szemléletesebb módon mutatja, hiszen a vizsgált frekvenciasávok határai logaritmikus skálán vannak egyenletesen elosztva. RMS² alapú kimenete van, melyet oszlopdiaagramon célszerű ábrázolni. Választható oktáv szélességgel kérdezhető le az eljárás (1, 1/3, 1/6, 1/12, és 1/24 oktávos). Vagy a lehető legnagyobb tömbre számítja a program az oktávanalízist, vagy a spektrumképzéshez hasonlóan a már említett átlagolási eljárással. A program szerkezete oly módon került kialakításra, hogy az számítás közben a részeredményeket a kezelőfelületen folyamatosan frissíti. Grafikusan vagy numerikusan állítható továbbá a vizsgált frekvenciatartomány, valamint megjelenítésre kerülnek a vizsgált jel és az oktáv analízis kimenetének elemszámai, terjedelme, valamint a kimeneti adatsor summája, a maximum érték és annak helye a frekvencia tengelyen. A megjelenítő grafikon nézete ez esetben is tengelyenként testre szabható.
- STFT – Short Time Fourier Transformation-t alkalmazó modul, mely a választott időjelre paramétereizhető módon számítja az STFT-t. A felhasználó 11 féle ablaktípusból választhat: None, Hanning, Hamming, Blackman-Harris, Exact Blackman, Blackman, Flat Top, 4 Term B-Harris, 7 Term B-Harris, Low Sidelobe vagy Gaussian. A megadott ablakméretet módosítást követően szoftveresen a legközelebbi 2 hatványra állítja a szoftver, ez a felhasználói felületre is kihat (a kontroll elem tartalma azonnal a számított 2 hatvány értékének megfelelően módosul. Választható alap felhasználói és haladó beállítási funkciók állnak rendelkezésre, valamint a megjelenítendő frekvenciartomány is paramétereizhető bemenetként adható meg. További paraméterként választható, hogy a felhasználó a számított STFT spektrumai helyett megjelenítheti az időszelethez tartozó frekvencia szerint normalizált, teljesítménysűrűségét is, így sokkal szemléletesebben megjelennek az időben változó domináns frekvenciakomponensek és a háromdimenziós megjelenítés skálázása is szemléletesebbé válik. További eredményekként megjelennek az időjel és a spektrogram főbb paraméterei (dt, elemszám, terjedelem; a spektrogram maximum értékének x, y és z tengelyen felvett értékei).

A megjelenítő grafikon megjelenítésének konfigurációja egy ikonra kattintva választható (szürkeárnyalatos, inverz szürkeárnyalatos vagy színes spektrogram).

A helyes beállítások megadását megkönnyíti, hogy a megjelenítő ablakban a statisztikai lekérdezések ablakához hasonlóan lejjebb görgetve megjelenik az ablakozás beállításának grafikus szemléltetése (grafikonon több, a választott ablaknak megfelelő formájú és elemszámú ablak görbéje rajzolódik ki, melyek egymáshoz képest a beállításoknak megfelelő mértékben vannak eltolva az átfedések szemléltetésének céljából).

Az eljárás folytonos algoritmussal valósul meg, mely lehetővé teszi kiterjedt adatsorok feldolgozását is. A program számítás közben a részeredményeket a kezelőfelületen folyamatosan frissíti. (11. ábra)

- 3D-s felületmegjelenítés – Az előzőekben ismertetett STFT eredménye külön ablakban egy 3D-s felületmegjelenítőn is megtekinthető. Egy erre alkalmas megjelenítőn lehet a 3D-s felületet a kurzorral forgatni, nagyítani, egyéb manipulációkat végezni rajta. (12. ábra)
- $RMS(\omega)^2$ – Teljesítményeloszlás-spektrumból számít frekvencia szerinti részintegrálok a megadott frekvenciatartományon belül. Így grafikusan szemlélteti a jel $RMS(\omega)^2$ függvényét, melynek végértéke az adott tartomány teljesítménytartalmát mutatja.
- RMSR – (RMS-Ratio – RMS arány) – A jelből számított teljesítményeloszlás-spektrum értékeit összegzi két, definiálható frekvenciatartományon belül, melyeket egy paraméterezhető osztási frekvencia választ el egymástól. A két tartományra számított RMS^2 értékek aránya sok esetben jól jellemzi az időjelet. Gyakran a magas frekvenciás komponensek esetén megnövekedett spektrumértékek valamilyen rendellenes mechanikai zajra, súrlódásra, kocogásra utalnak, így ez a számított arányszám jó alapja lehet némelyik összetettebb diagnosztikai eljárásnak is.
- $RMSR(\omega)$ – Az RMSR-hez hasonló algoritmus, mely egy adott frekvenciatartományon mutatja meg egy csúsztatott osztási frekvencia mellett kapott RMSR értékeket.
- Cepstrum – A rezgésdiagnosztikában elterjedt lekérdező algoritmus. Számítási módja, hogy a jelből képzett spektrum értékeinek logaritmusára elvégzünk egy Inverz Furier Transzformációt (IFT). 4 adatsort eredményez: komplex, valós, teljesítmény és fázis-cepstrumot. A teljesítmény cepstrumot használják elterjedten beszéd-elemzésre, de alkalmas lehet motordiagnosztikai megfigyelésekre is. [6]

IV. ÖSSZEFOGLALÁS

Az elkészült diagnosztikai program szerkezete eltérő az általában ilyen célra fejlesztett szoftverekétől. Praktikus, hogy egy specifikált, belső kiértékelési algoritmusokkal rendelkező adatgyűjtő rendszer egybe van integrálva a posztprocesszáló modullal, így egy programkörnyezeten belül lehet elvégezni komplett diagnosztikai vizsgálatokat. A közös keret további lehetősége, hogy a mérőprogram és az elemző között közvetlen információcsere valósulhat meg. Hátránya, hogy a fejlesztés a kezdeti stádiumban nehézkes, lassú, nagy hozzáértést kíván, de a benne rejlő új lehetőségek, a kényelmes és gyors kezelés és az elkészített keretrendszerben való funkcióbővítés és módosítás könnyebbsége teljes mértékben kompenzálja az imént említett nehézségeket.

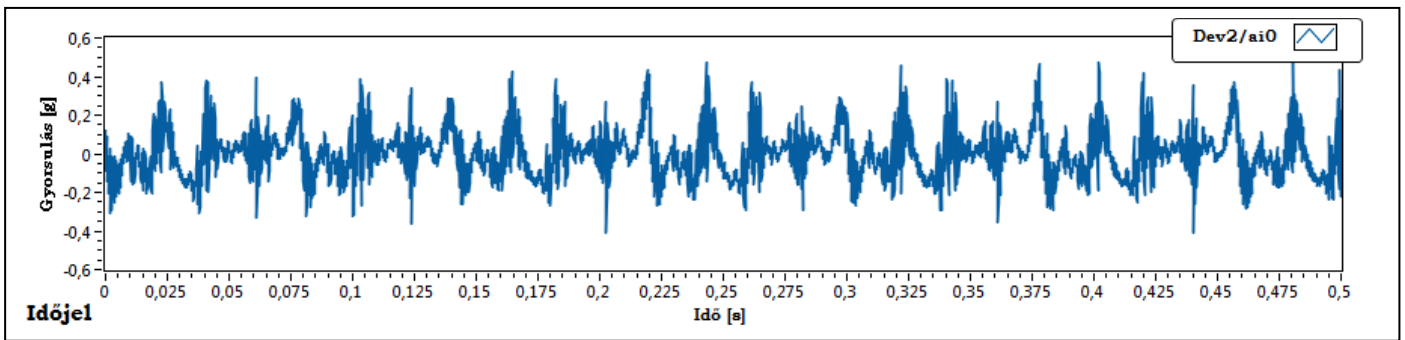
A multi-ablakos megjelenítő rendszer segítségével nagyon gyorsan és egyszerűen végezhető el egy adott mérési adatsor teljeskörű analízise, de egy adott módszer más paraméterekkel való lekérdezése és az eredmények együttes megjelenítése is megoldott. A gyors analízist több szinkronizációs funkció is segíti, mely beállításait minden ablakból elvégezhetjük, de külön megoldást kínál a keretrendszer egy központi ablakon keresztül a munkamenetek testre szabásához.

A szinkronizációs eljárás megvalósításához praktikus volt egy külön script nyelvet fejleszteni a műveletek kódolására, mely további lehetőségeket vet fel, úgy, mint a script alapú szöveges parancsbevitel, nem rendelkezésre álló adatsorok automatikus számítása a háttérben, sorozatelemzési lehetőségek, stb.

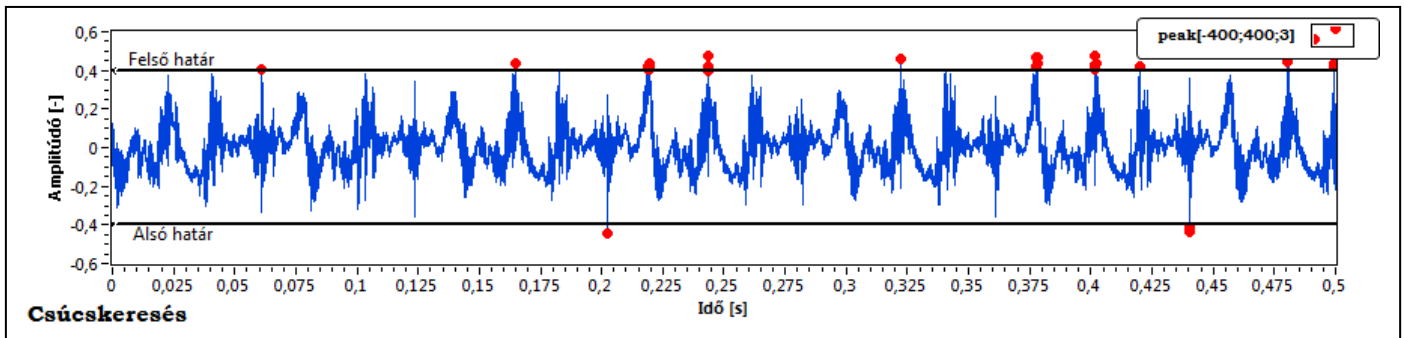
A multi-ablakos rendszer eredményeit egy külön adatkezelő ablak segítségével van mód a merevlemezre tárolni. A program csak azon lekérdező ablakok adatainak mentését ajánlja fel, melyek aktívak és tartalmazznak értelmezhető adatot. Az egész munkamenet összes adata (képek, adatsorok) egy gombnyomással mentésre kerül a strukturált mapparendszerben, de lehetőség nyílik szelektív mentésre is. A program mind a mérési adatok, a képfájlok és a letárolt adatfájlok tekintetében számos fájlformátumot támogat.

A szoftverstruktúra megtervezésekor kompromisszum nélkül lettek figyelembe véve az eddig diagnosztikai rendszerek hátrányai, a programot kezelő operátorok igényei, valamint a későbbi gyors és hatékony bővíthetőség.

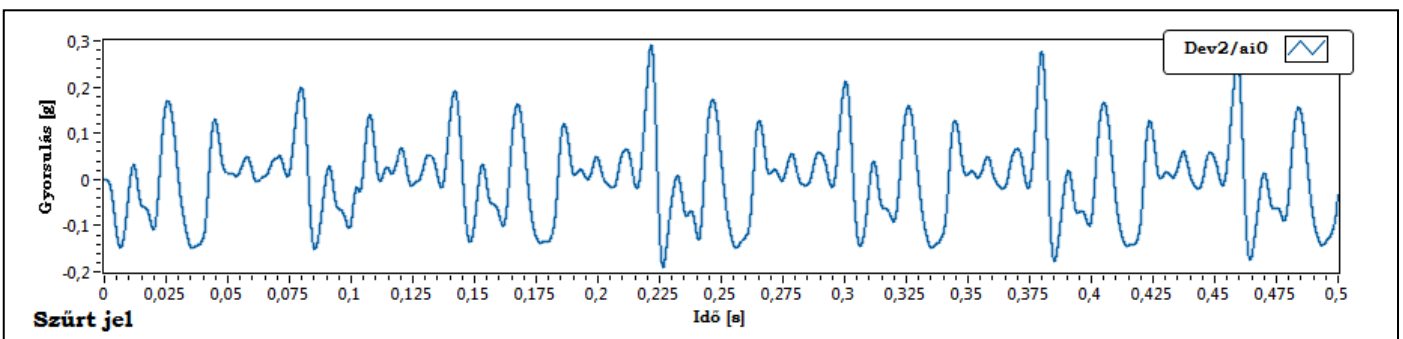
A keretrendszerbe a továbbiakban konkrét berendezésekre és célra specifikált algoritmusok kerülnek beépítésre, melyek segítségével már ipari diagnosztikai problémákat is kompakt módon meg lehet majd oldani. Az adatgyűjtő modul már jelenleg is számos mintavételező berendezést képes automatikusan felismerni, konfigurálni, kezelni. Továblépési lehetőség lehet FPGA alapú feldolgozás kifejlesztése, hogy valós idejű algoritmusok ne terheljék a számítógépet. A mintavételező – és így egyben a jelkondicionáló és feldolgozó egység – egy beágyazott rendszerként tudna számítógép-vezérelten, de attól függetlenül működni. A számítógépes felület ez esetben csak a kezelői és megjelenítő felületek funkcióját látná el.



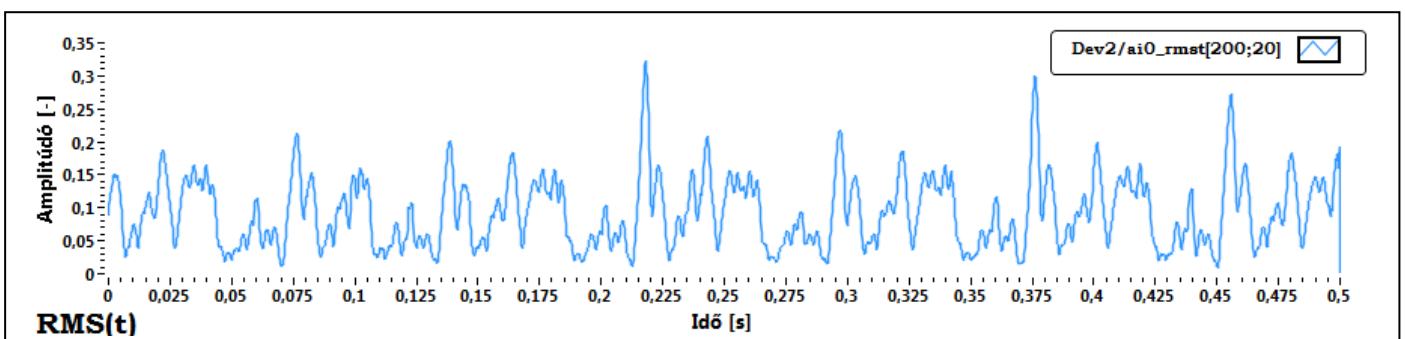
5. ábra: Időjel megjelenítése



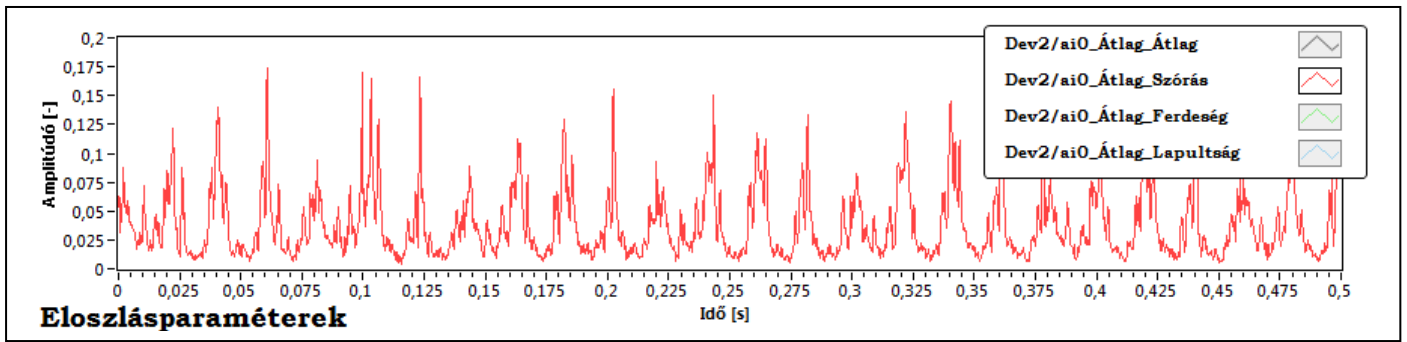
6. ábra: Csúskeresés eredményének megjelenítése a vizsgált időjellel együtt



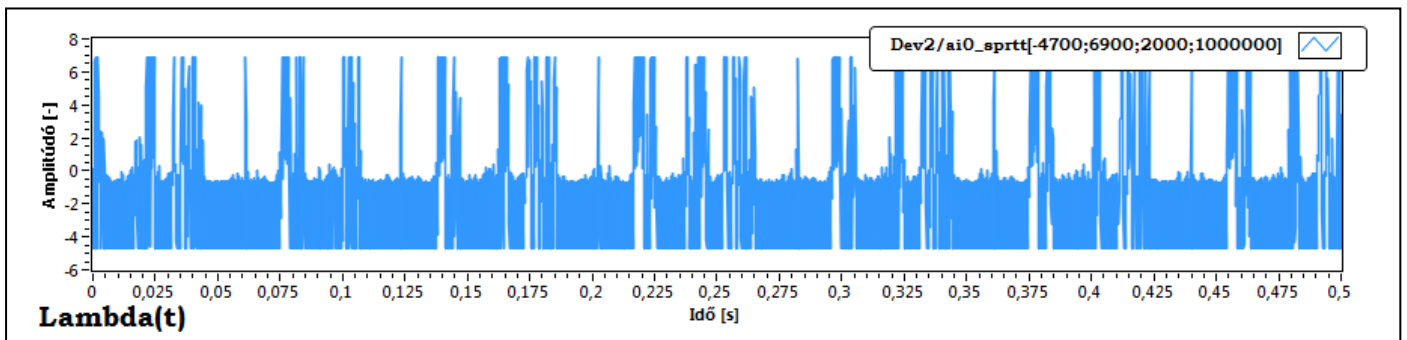
7. ábra: Aluláteresztő elliptikus szűrő alkalmazásának eredménye



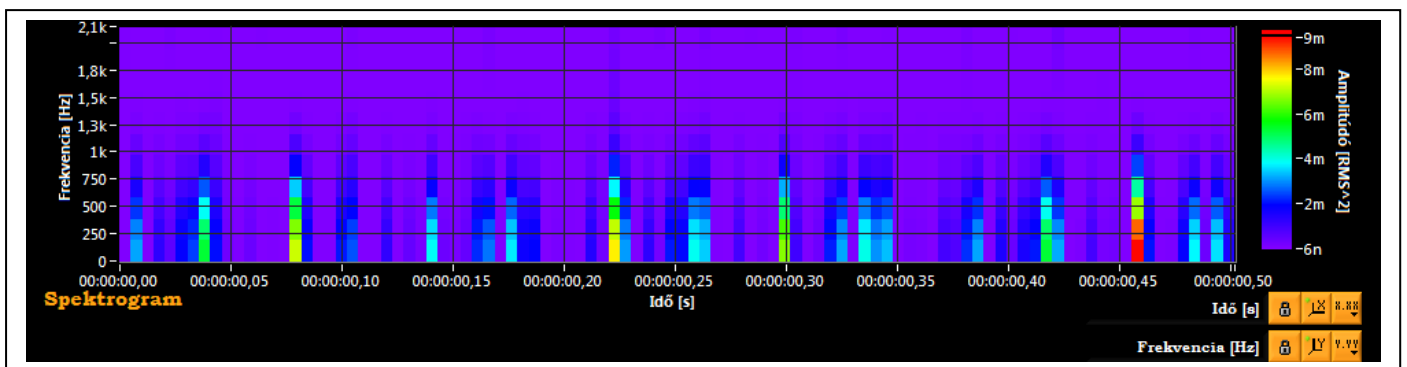
8. ábra: Csúsztatott ablakos RMS számítás eredménye



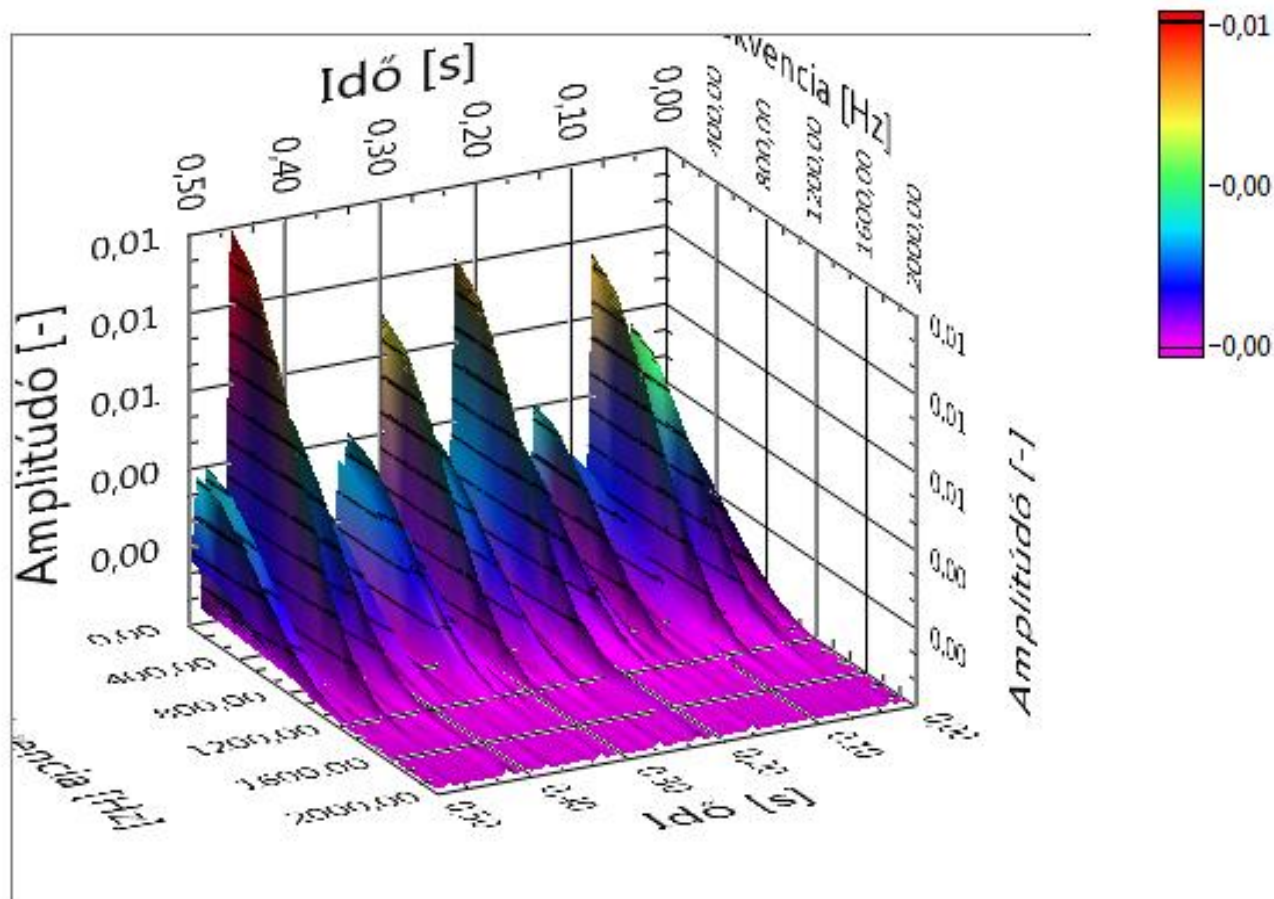
9. ábra: Eloszlás paraméterek számítása, majd választható jelek együttes megjelenítése (az ábrán csak a pillanatnyi szórás kerül megjelenítésre)



10. ábra: SPRT algoritmussal számított lambda függvény megjelenítése



11. ábra: STFT eredménye szintériképen ábrázolva



12. ábra: STFT eredményének 3D-s megjelenítése

Hivatkozások

- [1] J-D Wu, C-H Liu, "An expert system for fault diagnosis in internal combustion engines using wavelet packet transform and neural network", Expert Systems with Applications", vol. 36, pp. 4278-4286, 2009
- [2] J-D Wu, C-K Huang, "An engine fault diagnosis system using intake manifold pressure signal and Wigner-Ville distribution technique", Expert Systems with Applications, vol. 38, pp. 536-544, 2011
- [3] L Hunyadi, "About the fourth moment," Hungarian Statistical Review vol.87, pp. 262-286, 2009
- [4] WB Davenport, WL Root, "An Introduction to the Theory of Random Signals and Noise" Wiley-IEEE Press, New York, 1987
- [5] G. Manhertz, G. Gardonyi, G. Por (2014) Managing measured vibration data for malfunction detection of an assembled mechanical coupling, Int J Adv Manuf Technol [manuscript accepted for publication, DOI: 10.1007/s00170-014-6138-3]
- [6] H Konstantin-Hansen, H Herlufsen, "Envelope and Cepstrum Analyses for Machinery Fault Identification", Sound & Vibration, pp 10-12, May 2010