

Mérési útmutató az
Információfeldolgozás Laboratórium (VIMIM322)

Elosztott rendszerek és szenzorhálózatok

című méréséhez

Készítette:
Orosz György, Dr. Sujbert László
BME Méréstechnika és Információs Rendszerek Tanszék

2010. február

Tartalomjegyzék

5. mérés	2
5.1. A mérés célja	2
5.2. Szinkronizáció	3
5.2.1. Időbélyeg transzformáció megvalósítása	3
5.2.2. A mérőrendszer felépítése	5
5.3. Rezonátor alapú mérőrendszer tanulmányozása	8
5.3.1. Rezonátoros megfigyelő áttekintése	8
5.3.2. Szinkronizáció és jelrekonstrukció rezonátoros megfigyelőt használó rendszerben	12
5.3.3. A mérőrendszer felépítése	13
5.4. Rezonátor alapú aktív zajcsökkentő (ANC) rendszer tanulmányozása	17
5.4.1. Rezonátor alapú aktív zajcsökkentő rendszer bemutatása	17
5.4.2. A mérőrendszer felépítése	19
5.5. Felhasznált eszközök összefoglalása	23
5.6. Mérési feladatok	25

Ábrák jegyzéke

1. Lokális idők és szinkronizációs pontok	5
2. Az adatgyűjtést megvalósító mérőrendszer felépítése	6
3. Kezelői felület (adatgyűjtő)	7
4. A rezonátoros megfigyelő blokkvázlata	9
5. A Fourier-együtthatókból történő jelrekonstrukció időzítési diagramja	13
6. Kezelői felület (rezonátoros adatgyűjtés)	14
7. ANC blokkvázlata	17
8. ANC algoritmus magyarázata	19
9. ANC elrendezés (tesztelrendezés)	20
10. ANC elrendezés	20
11. ANC kezelői felület	21
12. Mitmót felépítése	23
13. DSP kártya	24
14. Analóg interface kártya	25

5. mérés

Elosztott rendszerek és szenzorhálózatok

5.1. A mérés célja

Manapság egyre több területen találkozhatunk különféle vezeték nélküli rendszerrel. Ezek a rendszerek igen nagy flexibilitást biztosítanak a vezeték nélküli adattovábbítás miatt, és általában különféle információgyűjtési és feldolgozási feladatot látnak el. Mivel a vezeték nélküli hálózatok egységei (a szenzor node-ok, vagy másképpen mótók) önálló működésűek, így a rendszer elosztott működésűnek tekinthető. Az elosztott rendszerek egyik nagy előnye, hogy az egyes egységek általában önálló vezérlőegységgel rendelkeznek, melyek akár lokális döntéseket képesek hozni, így teszik hatékonyabbá a rendszer működését.

A vezeték nélküli kommunikáció és az elosztott jelérzékelés és feldolgozás azonban több, alapvető problémát is felvet. A mérés célja, hogy megismertesse a hallgatókat az elosztott, vezeték nélküli jelfeldolgozó rendszerek jellegzetes jelfeldolgozási problémáival. A mérés során egy egyszerű, vezeték nélküli szenzorokra épülő rendszeren keresztül nyerünk betekintést különböző jelfeldolgozási problémákba. A vezeték nélküli hálózatot mótók alkotják, melyek egy, a 2.4 GHz-es sávban működő, CC2420 típusú kommunikációs IC-t használnak a rádiós adatátvitelhez.

A mérés három nagy témakör köré épül:

1. Szinkronizáció
2. Rezonátor alapú adattömörítés
3. Visszacsatolás megvalósítása vezeték nélküli szenzorhálózatban

Az első két feladat során a felhasznált szenzorhálózat egy PC-hez kapcsolódik, mely a szenzorok által gyűjtött adatok archiválására és feldolgozására szolgál, a harmadik részben pedig egy DSP (DSP: Digital Signal Processor: digitális jelfeldolgozó processzor) dolgozza fel valós időben (on-line módon) a szenzortól érkező jeleket.

A szinkronizáció problémája végigvonul a mérés során. Az első mérési feladatban egy egyszerű adatgyűjtő rendszer esetén ismerkedhetünk meg egy úgynevezett időbélyeg-transzformációt alkalmazó szinkronizációs technikával.

Második fő feladat egy rezonátor alapú megfigyelőt alkalmazó adattömörítést használó rendszer megismerése. Ebben az esetben egy vezeték nélküli szenzor előállítja a megfigyelt jel Fourier-együtthatóit, és csupán ezeket továbbítja a rádiós csatornán. Mivel ezek az együtthatók általában lassabban változnak mint maga a jel, így ritkábban is továbbíthatóak, ezzel valósítva meg az adattömörítést. A szinkronizáláson kívül ebben a rendszerben meg kell valósítani a jel visszaállítását a mért paraméterekből.

A mérés harmadik részében ezen rezonátoros adatgyűjtő rendszert használjuk fel egy aktív zajcsökkentő rendszerben. Ebben az esetben a vezeték nélküli rendszer egy DSP-hez kapcsolódik.

A mérés során a hallgatók gyakorlatot szerezhetnek jelfeldolgozási feladatok megvalósításában, és az elosztottan végzett megfigyelések értelmezésében, valamint a felmerülő hibák korrigálásában.

5.2. Szinkronizáció

5.2.1. Időbélyeg transzformáció megvalósítása

A szinkronizáció alapvető szerepet játszik az elosztott, vezeték nélküli rendszerekben. Ezekben az esetekben az egyes alegységek igen laza kapcsolatban állnak egymással a vezeték nélküli csatornán keresztül, emiatt autonóm egységeknek tekinthetők, melyek nem állnak *közvetlen* központi felügyelet alatt.

Tipikus és alapvető feladat a vezeték nélküli rendszerek esetén bizonyos analóg jelek (fényerő, hőmérséklet, hang ...) megfigyelése. A mérések kiértékelése azonban megfelelő időinformációt is feltételez a megfigyelések mellett. Például egy objektum haladási sebességének mérésekor nem elegendő csupán a helyzetének ismerete, hanem mindenképpen ismerni kell, hogy az objektum adott helyen milyen időpontokban tartózkodik. Mivel elosztott rendszerek esetén az egyes egységek általában önálló órával rendelkeznek, így az időinformációk szolgáltatása nem triviális feladat, mely főleg gyorsan változó jelek esetén jelent nagy kihívást. Ilyen jelek lehetnek például akusztikus jelek, melyeket gyakran használnak például különféle lokalizációs feladatok során.

Az időmérés alapjául általában valamilyen kvarcoszcillátor szolgál, mely igen jó kompromisszum az ár, pontosság és stabilitás szempontjából. (Ilyen kvarcoszcillátorok találhatóak például sok karórában is.) A nagyságrendek érzékeltetése szempontjából fontos tudni, hogy egy általános célú kvarc hibája maximum néhány tíz 10 ppm körül várható (ppm=parts per million= 10^{-6}). Ez azt jelenti, hogy pl. 10 ppm hiba esetén egy 10 MHz-es oszcillátor a névleges értéktől maximum ± 100 Hz-el tér el. Még ez a relatív kis hiba is jelentős eltérést okozhat az időmérésben hosszú idő alatt. Például, ha feltételezzük, hogy két szenzor órája a nap elején megegyezik, viszont a bennük található kvarcok frekvenciáinak eltérése 10 ppm, akkor nap végére az óráik által mutatott idő különbsége eléri a $24 \cdot 60 \cdot 60 \cdot 10 \cdot 10^{-6} = 0.864$ sec-t, tehát majdnem egy másodperces eltérés tapasztalható. Precíz mérések esetén ekkora hiba elfogadhatatlan.

A szenzorok szinkronizálására egy kézenfekvő módszer az órák folytonos hangolása (például a déli harangszó hallatára beállítjuk a karóráinkat pontosan délre). Az órák hangolásával történő (ún. fizikai) szinkronizációra láthatunk példát a Beágyazott és Ambiens Rendszerek Laboratórium 9-es mérésében (Villamosmérnöki alapszak, Beágyazott és irányító rendszerek szakirány, Beágyazott és ambiens rendszerek ágazat). Az órák hangolása azonban nem mindig lehetséges. Ebben az esetben használható például az időbélyeg-transzformációt megvalósító szinkronizációs algoritmus. Időbélyegnek nevezzük azt az időadatot, melyet valamilyen eseményhez rendelünk, pl. mikor történt egy megfigyelés.

Az algoritmus megismerése előtt néhány definíció tisztázása szükséges. A továbbiakban t jelöli a referencia időt, mely az abszolút pontos időt jelenti (pl. atomóra által szolgáltatott idő). $h_i(t)$ jelöli az i -edik node órája által mutatott időt. $O_{i,j}(t) = h_i(t) - h_j(t)$ az i -edik és a j -edik node órája közötti offset, tehát időkülönbség. $\rho_{i,j}$ az i -edik és a j -edik node órája közötti drift (elcsúszás), mely megadja, hogy a két óra közötti különbség mennyit változik egységnyi idő alatt: $\rho_{i,j} = \frac{dO_{i,j}}{dt} \approx (O_{i,j}(t_2) - O_{i,j}(t_1))/(t_2 - t_1)$. Ez utóbbi becslés akkor igaz, ha a két óra egyenletesen csúszik el egymáshoz képest, tehát a kvarcoszcillátorok frekvenciahibája állandó. Ez általában (legalább rövid távon) igaz, ekkor a drift megegyezik a kvarcok hibájával. Definiálhatjuk még az órák sebességét (gyorsaságát): $f_i = \frac{dh_i}{dt}$, mely azt mutatja meg, hogy milyen gyorsan telik az idő az i -edik node-on a referenciaidőhöz képest. Ideális esetben $f_i = 1$, tehát az óra egységnyi sebességgel jár. Az egymáshoz képest mért sebesség is definiálható hasonló módon: $f_{i,j} = \frac{dh_i}{dh_j}$.

Példa 1: Két óra ($mót_1$ és $mót_2$) által mutatott időpontokat feljegyeztük a következő időpontokban:

- [2000. jan. 10. 12:00:00]-kor
 - a $mót_1$ órája: [2000. jan. 10. 12:00:10]-t mutat

- a mót₂ órája: [2000. jan. 10. 12:00:09]-t mutat.
- [2000. jan. 15. 12:00:00]-kor
 - a mót₁ órája: [2000. jan. 15. 12:00:11]-t mutat
 - a mót₂ órája: [2000. jan. 15. 12:00:08]-t mutat.

Ebben a példában a referencia idő az első esetben: $t_1 = [2000. \text{ jan. } 10. \text{ 12:00:00}]$, tehát:

- $t_1 = [2000. \text{ jan. } 10. \text{ 12:00:00}]$ időpontban:
 - az első mót lokális ideje: $h_1(t_1) = [2000. \text{ jan. } 10. \text{ 12:00:10}]$
 - a második mót lokális ideje: $h_2(t_1) = [2000. \text{ jan. } 10. \text{ 12:00:9}]$
 - a két óra közötti ofszet tehát: $O_{1,2}(t_1) = h_1(t_1) - h_2(t_1) = 1 \text{ sec}$

A referencia idő a második esetben: $t_2 = [2000. \text{ jan. } 15. \text{ 12:00:00}]$, tehát:

- $t_2 = [2000. \text{ jan. } 15. \text{ 12:00:00}]$ időpontban:
 - az első mót lokális ideje: $h_1(t_2) = [2000. \text{ jan. } 15. \text{ 12:00:11}]$
 - a második mót lokális ideje: $h_2(t_2) = [2000. \text{ jan. } 15. \text{ 12:00:8}]$
 - a két óra közötti ofszet tehát: $O_{1,2}(t_2) = h_1(t_2) - h_2(t_2) = 3 \text{ sec}$.

A két időpont közötti időkülönbség: $\Delta t = t_2 - t_1 = 5 \text{ nap} = 432000 \text{ sec}$. A két mót órája közötti különbség tehát $(3-1) \text{ sec}$ -t változott Δt idő alatt, tehát a két óra közötti drift (elcsúszás sebessége): $\rho_{i,j} = (3-1) \text{ sec}/\Delta t = 4.63 \cdot 10^{-6} = 4.63 \text{ ppm}$.

Itt jegyezzük meg, hogy az óráknak nem fontos a hagyományos értelemben vett időt és dátumot mutatni, lehet ez az időinformáció például a bekapcsolásuktól számított idő, mint ahogyan az a mérésben használt rendszer esetében látni fogjuk.

Folytatva a példát, tegyük fel, hogy a két óra egy híd két végén van elhelyezve, és méri, hogy egy adott objektum mennyi idő alatt halad át a hídon. Az 1. óra azt jelentette, hogy egy autó $h_1(t_3) = [2000. \text{ jan. } 25. \text{ 12:00:13}]$ -kor hajtott rá a hídra a 2. óra pedig azt jelentette, hogy az adott autó $h_2(t_4) = [2000. \text{ jan. } 25. \text{ 12:00:11}]$ -kor hajtott le. Mennyi idő alatt hajtott át az autó?

A megoldáshoz közös időskálára kell hoznunk a két órát, tehát el kell végeznünk egy időbélyeg-transzformációt. Az előző számításokból tudjuk, hogy $[2000. \text{ jan. } 15. \text{ 12:00:00}]$ -kor az órák közötti ofszet $O_{1,2} = 3 \text{ sec}$ valamint láthatjuk, hogy az ofszet 5 nap alatt 2 sec -t változik, tehát $[2000. \text{ jan. } 25. \text{ 12:00:13}]$ -kor az ofszet kb. $3 \text{ sec} + 2 * 2 \text{ sec} = 7 \text{ sec}$. Ezek szerint, ha az első óra időskáláját vesszük alapul, akkor megállapíthatjuk, hogy az autó az első óra szerint $h_1(t_4) = h_2(t_4) + 7 \text{ sec} = [2000. \text{ jan. } 25. \text{ 12:00:18}]$ -kor hajtott le a hídról, ami azt jelenti, hogy az autó $h_1(t_4) - h_1(t_3) = 5 \text{ sec}$ alatt haladt át a hídon. (Megjegyezzük, hogy itt tettünk néhány ésszerű egyszerűsítést: pl. nem vesszük figyelembe, azt, hogy a két óra valamennyit elcsúszik egymáshoz képest annyi idő alatt is, míg az autó áthalad a hídon, de ez ebben az esetben nagyságrendi hibát nem okoz.) Láthatjuk, hogy mivel az órák nem ideálisak, így egymástól függetlenül járnak, tehát nem lehetséges az általuk mutatott időadatokat (időbélyegeket) közvetlen felhasználása, hanem valamilyen szintű átalakítás, tehát időbélyeg-transzformáció szükséges. A következő részben ennek a transzformációnak a matematikai alapjai kerülnek bemutatásra, melyek felhasználásával egyszerű módon végrehajtható a transzformáció.

Általánosságban elmondható, hogy két egység esetén az egyes egységek lokális idejei felírhatóak egymás függvényeként:

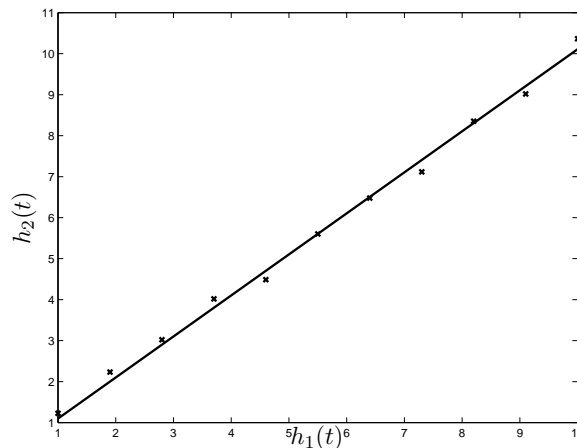
$$h(t)_i = g(h_j(t)). \quad (1)$$

Egy gyakorlatban is jól használható összefüggést kapunk, ha feltételezzük, hogy a lokális idők egymásnak lineáris függvényei, ekkor:

$$h_i(t) = ah_j(t) + b, \quad (2)$$

A lineáris függvény azt jelenti, hogy azt vesszük figyelembe, hogy az órák kvarcoszcillátorainak frekvenciahibája állandó. Ez általában, legalábbis rövid távon igaz. Amennyiben feltételezzük, hogy (2) igaz, és ismerjük az a és b paramétereket, akkor az i -edik és j -edik node órája között az átváltást egyértelműen elvégezhetjük. Ezt a műveletet nevezzük időbélyeg-transzformációnak. Ezzel elérhetjük, hogy konziszens időeredményeket kapjunk és két, egymástól különálló szenzor által érzékelt eseményeket értelmezni tudjuk. Természetesen több egység esetén is alkalmazható a módszer, ebben az esetben ki kell választani, hogy mi legyen a közös időskála.

(2) egyenletben található paraméterek meghatározhatóak úgynevezett szinkronizációs pontok segítségével, melyek ugyanazon időponthoz tartozó időbélyeg párok. A példánkban ezek az időbélyeg párok: $\{h_1(t_1), h_2(t_1)\}$ valamint $\{h_1(t_2), h_2(t_2)\}$. Két szinkronizációs pont alapján már meghatározható (2) egyenlet két paramétere: a és b . A paraméterek meghatározásához általában több szinkronizációs pontot használunk, így a mérési hibák hatása csökken. Elsőfokú egyenlet esetén, mint például (2), a paraméterek meghatározásához használható a lineáris regresszió. Erre látható példa az 1. ábrán. A MATLAB hatékony segítséget nyújt a különböző fokú görbeillesztéshez a `polyfit` parancs segítségével.



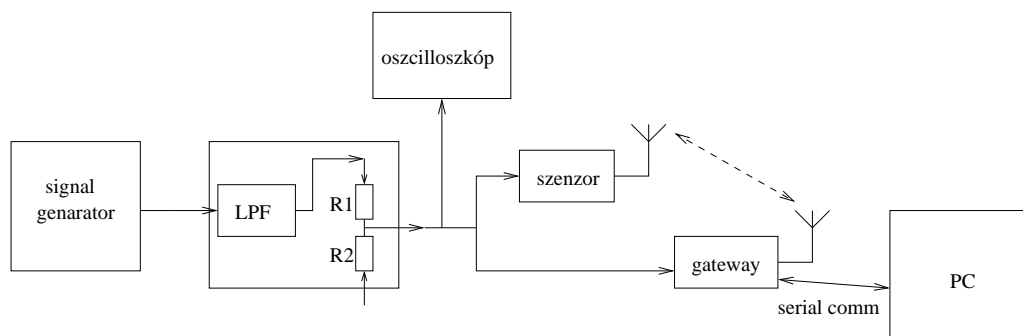
1. ábra. Lokális idők függvényei és szinkronizációs pontok. Folytonos vonal: $h_2(t) = ah_1(t) + b$, \times : $\{h_1(t_k), h_2(t_k)\}$ szinkronizációs pontok.

Általánosságban tehát elmondható, hogy az időbélyeg transzformáció elvégzése három lépésben hajtható végre:

1. Szinkronizációs pontok gyűjtése.
2. A szinkronizációs pontok alapján meghatározzuk az időbélyegek átváltásához szükséges paramétereket: a és b a (2)-es egyenletben.
3. Az időbélyegek konverziója az előző pontban meghatározott paraméterek és (2) alapján.

5.2.2. A mérőrendszer felépítése

A mérés első részében a 2. ábrán látható elrendezést használjuk.



2. ábra. A mérőrendszer felépítése az adatgyűjtést megvalósító rendszer esetén (mintavételező és rezonátoros megfigyelőt alkalmazó szenzor esetén is)

Ebben a rendszerben két módot találhatók (lásd: 12. ábra). Az egyik pusztán szenzor funkcióval rendelkező módot (továbbiakban szenzor). Ez a módot az AD átalakítójára vezetett jelet mintavételezi és rádióon keresztül továbbítja az átjáróként funkcionáló módot felé. Az átjáró (gateway) módot fogadja az adatokat, és soros porton továbbítja a PC felé. Az átjáró módot ezen kívül szenzor funkciót is ellát, és a szenzorhoz hasonlóan mintavételezi az AD átalakítójára vezetett jelet és a saját adatait is küldi a PC felé soros porton. A mérés első részében ugyanazt a jelet vezetjük a szenzor és az átjáró bemenetére, így ellenőrizhetjük a szinkronizáció pontosságát: mindkét módot esetén ugyanazt a jelet kell érzékelni eltekintve az analóg elektronika miatti különbségektől. Külső jelként egy függvénygenerátor jelet használjuk. Ezt a jelet egy egyszerű aluláteresztő szűrőre (LPF: Low Pass Filter) vezetjük, mely sávkorlátozó szűrőként funkcionál, így csökkentve a véges mintavételi frekvencia miatti átlapolódás hatását. (R1 és R2 ellenállások szerepét lásd később, ebben a mérésben nincs hatásuk.) A szűrő a 14. ábrán látható.

Későbbiekben látjuk majd, hogy a szenzor kétféle üzemmódban is képes működni (mintavételezés vagy előfeldolgozás). Az üzemmódot bekapcsolás után SW1 és SW2 nyomógombokkal lépkedve 1-es módba kell állítani (a 7 szegmenses kijelzőn láthatjuk a sorszámot). Választás után az SW3 gomb hosszú megnyomásával nyugtázhatjuk a kiválasztott üzemmódot.

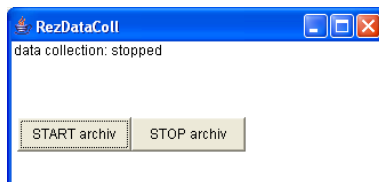
A módotokon a mintavételi frekvencia névleges értéke megegyezik, mégpedig, 1800.2 Hz, egészen pontosan: $f_s = (8 \cdot 10^6 / 4444)$ Hz.

Az adatok továbbítása csomag alapú, mind a szenzor mind az átjáró adatai 25 mintát tartalmazó csomagokban kerülnek továbbításra a PC felé. Minden csomag el van látva egy időbélyeggel, ami a csomag első mintájának mintavételezési időpontját mutatja. Ez az időpont az adott módot bekapcsolása óta eltelt időt adja meg.

Az átjáró módot az adatokon kívül továbbítja a PC felé a szinkronizációs pontokat, melyek segítségével az időbélyeg transzformációt elvégezhetjük. A szinkronizációs pontok kialakítása a rádiós kommunikáció segítségével történik kihasználva a rádiós IC (CC2420) szolgáltatásait. A CC2420-as IC egy speciális funkciója, hogy adás kezdetekor az adó és vevő oldalon egy digitális kimeneten (SFD láb: Start of Frame Delimiter) egy felfutó él generál az IC (részletekért lásd az adatlapot). Ez a felfutó él igen nagy pontossággal azonos időpontban generálódik az adó és vevő oldalon, tehát ha mind az adó mind a vevő feljegyzi ezt az időpontot, akkor az így kialakuló időbélyegpárok felhasználhatóak szinkronizációs pontokként. Az adott adáshoz tartozó időpontot az adó még az aktuális csomagban elhelyezi, így a vevő a saját maga által feljegyzett időbélyeggel együtt továbbítani tudja a PC felé, ahol ezeket a szinkronizációs pontokat felhasználhatjuk az időbélyeg-transzformáció végrehajtásához.

Az átjáró módot által küldött adatokat egy PC-s program segítségével menthetjük el a me-

revlemezre. Az elmentett file-okat a mérésen rendelkezésre álló `preprocessSampleFiles.m` file meghívásával egyszerűen feldolgozható formátumúra alakíthatjuk.



3. ábra. Az egyszerű mintavételezést alkalmazó mérőrendszer kezelői felülete.

Az előfeldolgozás után a következő file-okat használhatjuk a mérés során:

pre_stmpSync.dat: a szinkronizációs pontpárokat tartalmazó file. A file két oszlopból áll, minden sorban egy-egy szinkronizációs pontpár található. Az első oszlopban található az átjáró a második oszlopban a szenzor által generált időpontok, melyek ugyanahhoz a rádiós adáshoz tartoznak. Például egy lehetséges eredmény:

```
4068.618299375 3.014941125
4068.632187125 3.028829000
4068.646074750 3.042716625
4068.659961625 3.056603625
4068.673849375 3.070491625
4068.687732125 3.084374500
4068.701622875 3.098265250
4068.715510625 3.112153125
4068.729398375 3.126041125
```

Tehát az első rádiós adás az átjáró bekapcsolása után 4068.618299375 sec-al a szenzor bekapcsolása után pedig 3.014941125 sec-al történt, és ez a két időpont egyazon eseményhez tartozik.

A file-t (és általában a többit is) a MATLAB `load` parancsával egyszerűen beolvashatjuk, például:

```
readSync = load('pre_stmpSync.dat');
```

egy két oszlopot tartalmazó vektort ad vissza, mely a file tartalmával egyezik meg.

pre_gwySampFull.dat, **pre_sensorDataFull.dat**: a gateway és a szenzor által küldött adatokat és a hozzájuk tartozó időbélyegeket tartalmazza. A második oszlopban a minták az első oszlopban az adott mintához tartozó időbélyeg található.

5.3. Rezonátor alapú mérőrendszer tanulmányozása

A mérés második részében egy rezonátor alapú tömörítést alkalmazó mérőrendszer vizsgálatára kerül sor. A mérőrendszer fizikai felépítése megegyezik a 2. ábrán látható elrendezéssel, tehát egy szenzor, valamint egy átjáró és szenzor funkciót is ellátó mót mintavételezi ugyanazon jelet. A különbség a két rendszerben az, hogy míg az első esetben a szenzor a jel mintavételezését és a minták továbbítását végezte, ezen mérés során a szenzor előállítja a jel Fourier-együtthatóit, és az együtthatókat továbbítja. Ez a rendszer periodikus jelek esetén alkalmazható, mivel a Fourier-együtthatók segítségével periodikus jelek állíthatóak elő. A rendszer előnye, hogy periodikus jelek esetén csökkenthető a rádiós hálózaton átküldendő adatok mennyisége, ugyanis a periodikus jelek Fourier-együtthatói általában lassabban változnak mint maga a jel. Esetünkben a szenzor feleannyi adatot továbbít, mint az egyszerű mintavételező szenzor esetében.

A rendszer demonstrálja az intelligens szenzorok által kínált lehetőséget, nevezetesen a szenzor már helyben alkalmas bizonyos feladatok elvégzésére. Ez a számítási teljesítmény felhasználható a mért jelek előfeldolgozására, mely segíthet a központi egység tehermentesítésében, vagy amint esetünkben is tapasztalható, a kommunikációs teher csökkentésében, hiszen elegendő a mért paramétereket továbbítani. A következő fejezetekben megismerkedhetünk a rezonátoros megfigyelő felépítésével, mely a rendszer egyik alapvető építőeleme.

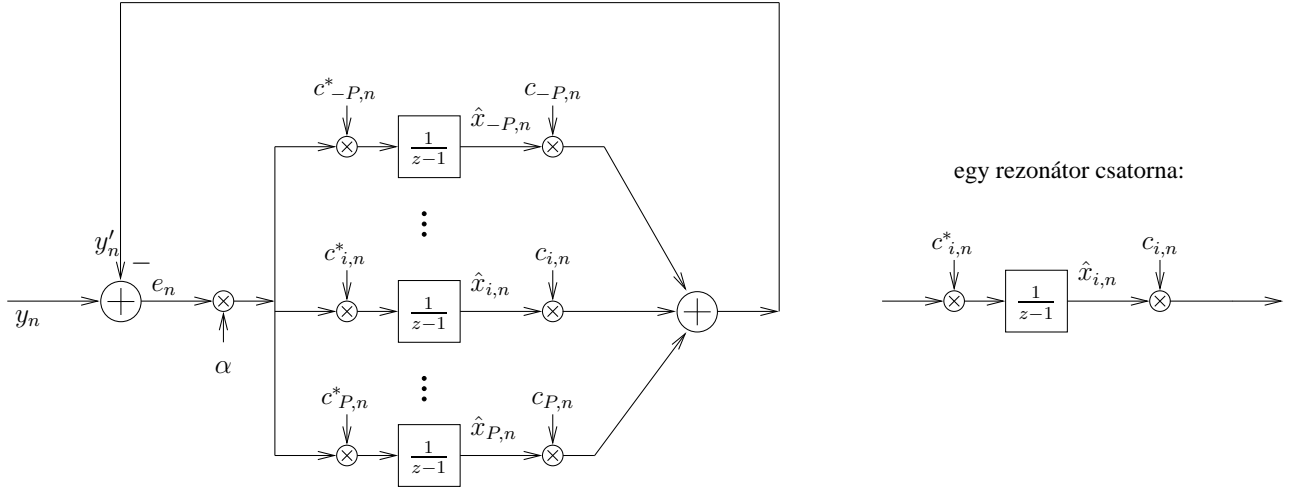
5.3.1. Rezonátoros megfigyelő áttekintése

A hallgatók a Méréselmélet (vimim108) című tárgy keretei között részletesen megismerkedhetnek a rezonátoros megfigyelő algoritmussal (továbbiakban RBO: Resonator Based Observer). Habár a mérés során nem szükséges az algoritmus részletes ismerete, csupán „alkalmazói” szinten használjuk, egy rövid áttekintést adunk az algoritmusról azért, hogy betekintést kapjunk a rendszer működésébe.

A rezonátor alapú megfigyelő periodikus jelek Fourier-együtthatóinak rekurzív számítására szolgál. Az algoritmus egyik potenciális alternatívája lehet a hagyományos frekvenciatartománybeli diszkrét Fourier-transzformációknak (DFT illetve FFT), azonban a két algoritmus a következő alapvető eltéréseket mutatja:

1. A DFT az egész frekvenciatengely mentén egyenletesen kiszámítja a jel spektrumát, míg az RBO alkalmas csupán a periodikus jelek Fourier-együtthatóinak számítására. Ez jelentős erőforrás megtakarítást jelent, mivel nem kell a spektrumot olyan helyeken is számítani, ahol nem található hasznos komponens.
2. Az RBO-hoz szükséges a jel frekvenciájának ismerete, míg a DFT-nél nem. Amennyiben ismert a jel frekvenciája, az RBO kiküszöböli a DFT esetén fellépő tetőesés (picket fence) és szivárgás (leakage) jelenségeket.
3. A BRO rekurzívan állítja elő a Fourier-együtthatókat, tehát mintáról-mintára frissíti a becslést. Ezzel szemben a DFT blokkos feldolgozást végez, tehát a transzformációhoz szükséges összes adatot össze kell gyűjteni a transzformáció elvégzése előtt, mely adott pontosság esetén általában nagyobb késleltetést visz a rendszerbe.

Az RBO algoritmus megismerése előtt bemutatjuk a periodikus jelek felírásának módját, mely azért szükséges, mivel az RBO periodikus jelek paramétereit méri. Legyen adott egy y_n periodikus jel, mely P számú harmonikust tartalmaz, a mintavételi frekvenciához viszonyított relatív frekvenciája Θ , az i -edik harmonikushoz tartozó Fourier-együtthatói az n időpontban



4. ábra. A rezonátoros megfigyelő blokkvázlata

pedig $x_{i,n}$. Ekkor az y_n jelet a következő alakban írhatjuk fel:

$$y_n = \sum_{i=-P}^P x_{i,n} e^{j\Theta i n} = \sum_{i=-P}^P x_{i,n} c_{i,n}, \quad \text{és } c_{i,n} = e^{j\Theta i n} \quad (3)$$

ahol n az időindex, $c_{i,n} = e^{j\Theta i n}$ pedig az i -edik felharmonikushoz tartozó bázisfüggvény. A fenti egyenlet általános mintavételes rendszerekben érvényes. Amennyiben valódi időskálát használunk, akkor a következő érvényes:

$$y(t) = \sum_{i=-P}^P x_i(t) e^{j f i t} = \sum_{i=-P}^P x_i(t) c_i(t). \quad (4)$$

Ebben az esetben f a frekvenciát jelöli, így f_s mintavételi frekvenciát feltételezve: $\Theta = f/f_s$. A kétféle leírási mód –(3) és (4)– azért szükséges, mert a mérés során elosztott működésű rendszerekkel foglalkozunk, ahol általánosságban nem beszélhetünk globális mintavételi időpontokról, viszont az egyes egységeken belül már lokális mintavételi időpontokról igen. Így pl. az egyes egységeken futó algoritmusok leírásánál jobban használható (3), viszont az egymás között cserélt adatok értelmezése csak egy abszolút, közös időskálával lehetséges, mint ahogyan (4) esetén látjuk.

(3) egyenletnek létezik egy tömörebb formája is, mely a felírást egyszerűsíti. Ebben felhasználjuk a következő jelöléseket:

$$\begin{aligned} \mathbf{x}_n &= [x_{-P,n} \quad \dots \quad x_{i,n} \quad \dots \quad x_{P,n}]^T & : \text{ oszlopvektor} \\ \mathbf{c}_n &= [c_{-P,n} \quad \dots \quad c_{i,n} \quad \dots \quad c_{P,n}]^T & : \text{ oszlopvektor,} \end{aligned} \quad (5)$$

ahol \mathbf{c}_n a bázisfüggvényeket tartalmazó vektor az n -edik időpontban \mathbf{x}_n pedig a Fourier-együtthatókat tartalmazó vektor az n -edik időpontban. Ezekkel a jelölésekkel, felhasználva a vektorszorzás szabályait¹, (3) a következő egyszerű formában írható:

$$y_n = \mathbf{c}_n^T \mathbf{x}_n. \quad (6)$$

¹ $\mathbf{a}^T \mathbf{b} = \sum_i a_i b_i$

Az RBO az y_n jelnek az \mathbf{x}_n Fourier-együtthatóit becsli. Az algoritmus felépítése a 4. ábrán látható. Jelöljük az \mathbf{x}_n Fourier-együtthatók becslőit tartalmazó vektort $\hat{\mathbf{x}}_n$ -nel. A megfigyelőt a következő egyenletek írják le:

$$y'_n = \mathbf{c}_n^T \hat{\mathbf{x}}_n, \quad (7)$$

$$e_n = y_n - y'_n = y_n - \mathbf{c}_n^T \hat{\mathbf{x}}_n, \quad (8)$$

$$\hat{\mathbf{x}}_{n+1} = \hat{\mathbf{x}}_n + \alpha e_n \mathbf{c}_n^*. \quad (9)$$

(Megjegyezzük, hogy létezik ettől eltérő felírási mód is. Alkalmazástól függ, hogy melyik felírás praktikusabb, de a felírási mód az alapvető tulajdonságokat nem befolyásolja.) * a komplex konjugálást jelöli. α egy ún. bátorsági tényező, ennek meghatározására nem térünk ki. Kvalitatíve annyi mondható, hogy kis értékű α jó zavarelnyomást eredményez a mérési zajokkal szemben, nagy értékű α gyors beállást biztosít; a konkrét választás kompromisszum eredménye. Látható, hogy az algoritmus követi a rekurzív becslési algoritmusok szokásos alakját:

1. A rendelkezésre álló $\hat{\mathbf{x}}_n$ becsült paraméterek alapján becslést ad az y_n jelre. Lásd (7) egyenletben: y' .
2. Kiszámítja a becslés hibáját: (8).
3. Az e_n becslési hiba alapján módosítja a becsült paramétereket: (9).

Az algoritmus többféleképpen értelmezhető. A következő pontokban felsorolunk néhány értelmezési módot, megkönnyítendő az algoritmus megismerését/megértését:

1. Hagyományos felfogás szerint abból indulunk ki, hogy az y_n jelet úgy tekintjük, mint egy lineáris rendszer kimenetét, melynek állapotegyenletei:

$$\begin{aligned} \mathbf{x}_{n+1} &= \mathbf{x}_n \\ y_n &= \mathbf{c}_n^T \mathbf{x}_n. \end{aligned} \quad (10)$$

Tekinthetjük (10)-t szemléletesen pl. egy függvénygenerátort, mint fizikai rendszert leíró állapotegyenletnek. Ezen lineáris rendszer állapotváltozói a megfigyelt jel Fourier-együtthatói, tehát az ehhez a rendszerhez tervezett állapotmegfigyelő (RBO) a jel Fourier-együtthatóit állítja elő.

2. Az algoritmus formailag nagyban hasonlít az LMS algoritmusához (lásd Információfeldolgozás laboratórium 3. mérés), és valóban értelmezhető egy komplex értékű LMS algoritmusként. Ebben az esetben \mathbf{c}_n felfogható úgy mint referenciajel, cél pedig, hogy a \mathbf{c}_n bázisfüggvényekkel négyzetes értelemben minél jobban közelítsük az y_n jelet, tehát minimalizáljuk a következő költségfüggvényt: $J(n) = |e_n|^2 = |y_n - \mathbf{c}_n^T \hat{\mathbf{x}}_n|^2$. $\hat{\mathbf{x}}_n$ paramétervektort minden időpontban a költségfüggvény negatív gradienseinek irányába módosítjuk egy α bátorsági paramétert használva. A negatív gradiens (figyelembe véve, hogy $\hat{\mathbf{x}}_n$ komplex szám): $-\nabla J(n) = -\frac{\partial J(n)}{\partial \hat{\mathbf{x}}_n} = \mathbf{c}_n^* (y_n - \mathbf{c}_n^T \hat{\mathbf{x}}_n) = \mathbf{c}_n^* e_n$. Tehát $\hat{\mathbf{x}}_{n+1} = \hat{\mathbf{x}}_n - \alpha \nabla J(n)$ a (9) egyenletet szolgáltatja.
3. Az egyes rezonátorcsatornák az adott rezonátor frekvencián végtelen átvitelrel rendelkeznek, tehát felfoghatóak úgy, mint a rezonátor frekvenciákra eltolt integrátor: a $c_{i,n}^*$ -al való szorzás eltolja a i -edik rezonátor frekvencián található jelet DC-re, ahol integráljuk, majd $c_{i,n}$ -al való szorzás visszatolja az integrált jelet az eredeti frekvenciára. Szabályozástechnikából ismert, hogy egy visszacsatolt integrátor hiba nélkül tud követni konstans jelet, így a rezonátor frekvenciákra eltolt végtelen erősítés segítségével hiba nélkül előállítható minden, a rezonátor frekvenciákon található periodikus jel is.

A konkrét rendszerben a szenzor maximum 5 darab Fourier-együttható számítására képes. Ez az érték a mikrokontroller (ATmega128 @ 8 MHz) számítási teljesítménye által korlátozott. A mót konfigurálható úgy is, hogy csupán a páratlan Fourier-együtthatókat számítsa. Ez azokban az esetekben hasznos, ha a jel csupán páratlan harmonikus pozícióban található Fourier-együtthatókat tartalmaz (pl.: háromszög jel, négyszög jel, vagy általában a villamos hálózat feszültség/áramfüggvénye). Ebben az esetben a páros harmonikusok számítása felesleges számítási teljesítményt kötne le. A DC komponenst a mót nem számítja, mivel a móthoz tartozó interfész kártya AC csatolású. A szenzor mótot az IO kártyán található 1-es kapcsolóval lehet konfigurálni, hogy az első öt harmonikust (ON) vagy az első öt *páratlan* harmonikust számítsa és továbbítsa (OFF).

Mivel a megfigyelt y_n jel valós értékű, így teljesül a következő egyenlőség:

$$x_{i,n} = x_{-i,n}^* \quad (11)$$

tehát a negatív frekvenciákon található Fourier-együttható a pozitív frekvenciákon található komponensek konjugáltjai, emiatt a szenzor csupán $\{x_{i,n}; i > 0\}$ komponenseket továbbítja, hiszen y_n valós és DC mentes jel egyértelműen előállítható ezen adatokból is:

$$y_n = 2\Re \left\{ \sum_{i=1}^P x_{i,n} c_{i,n} \right\}, \quad (12)$$

ahol $\Re\{\cdot\}$ a valós rész képzést jelöli. (12) egyszerűen megkapható (3) és (11) egyenletekből, felhasználva, hogy egy γ komplex szám esetén²: $\gamma + \gamma^* = 2\Re\{\gamma\}$

Frekvenciamérés rezonátoros megfigyelővel. A megfigyelő megfelelő működéséhez meg kell adni a jel frekvenciáját. Erre mérés során egy PC-s kezelői felület szolgál (lásd később a 6. ábrán). A frekvencia pontos mérése a szenzor mérési eredményei alapján történik, mégpedig az Adaptív Fourier Analizátor (AFA) alap gondolatát felhasználva. Ebben a következő összefüggést alkalmazzuk. Legyen adott egy ω_{jel} frekvenciájú és ϕ_{jel} fázisú jel: $y(t) = \cos(\omega_{jel}t + \phi_{jel})$. Amennyiben az $y(t)$ jelet egy ω_1 frekvenciájú bázisfüggvény segítségével állítjuk elő, akkor a következőt kapjuk: $y(t) = \cos[\omega_{jel}t + \phi_{jel}] = \cos\{\omega_1t + [(\omega_{jel} - \omega_1)t + \phi_{jel}]\}$, amely:

$$y(t) = \cos\{\omega_1t + [(\omega_{jel} - \omega_1)t + \phi_{jel}]\} = \cos\{\omega_1t + \phi'_{jel}\}, \quad (13)$$

tehát egy olyan, ω_1 frekvenciájú jelet kapunk, melynek fázisa a két frekvencia különbségével arányosan változik az időben, tehát:

$$\phi'_{jel} = (\omega_{jel} - \omega_1)t + \phi_{jel} = 2\pi(f_{jel} - f_1)t + \phi_{jel}. \quad (14)$$

Amennyiben a rezonátorok frekvenciáját beállítjuk egy becsült frekvenciára, akkor a jel frekvenciájának a beállított frekvenciától való eltérését az mutatja, hogy a megfigyelt jel fázisa milyen gyorsan változik (ez szemléletesen azt jelenti, hogy a megfigyelt jel Fourier-együtthatója a frekvenciaeltérés arányában forog). Amennyiben ábrázoljuk a fázist, a fázisra illesztett egyenes meredeksége megadja a frekvenciaeltérést (a meredekség $\frac{\text{rad}}{\text{sec}}$ -ben értelmezendő, tehát osztani kell 2π -vel, hogy frekvenciát kapjunk). Egy komplex szám fázisa a MATLAB `angle` parancsával lehetséges, ezen kívül hasznos az `unwrap` parancs használata is (részletekért lásd MATLAB: `help unwrap`).

Manuálisan úgy kereshetjük meg a frekvenciát, hogy az alapharmonikus Fourier-együttható fázisát figyeljük (a rendszerhez tartozó kezelői felületen látható: 6. ábra), és amennyiben az

²Amennyiben $\gamma = a + jb$, akkor $\gamma^* = a - jb$, így: $\gamma + \gamma^* = a + jb + a - jb = 2a$, tehát γ valós részének kétszerese.

óramutató járásával ellentétes irányba forog, akkor a frekvenciát növelni kell, ellentétes esetben csökkenteni kell. A forgás sebessége arányos a frekvenciahibával: ha a fázis $d\phi$ fokot változik két egymást követő Fourier-együttható érkezése között, mely $T_F \approx \frac{1}{36}$ sec, akkor a frekvenciaeltérés: $\frac{d\phi}{360^\circ \cdot T_F}$. $d\phi$ értéke közvetlenül leolvasható a rendszerhez tartozó kezelői felületről: 6. ábrán dFi.

5.3.2. Szinkronizáció és jelrekonstrukció rezonátoros megfigyelőt használó rendszerben

A mérés során célunk, hogy a szenzor által előállított, majd az átjáró móton keresztül a PC-hez továbbított Fourier-együtthatókból visszaállítsuk a mért jelet. A visszaállított jel összehasonlítható az átjáró mótt által mintavételezett adatokkal, így tesztelhetjük az algoritmus tulajdonságait.

Fontos megjegyezni, hogy a szenzor mótt időskáláját ebben a rendszerben is transzformálni kell, ehhez rendelkezésre állnak a szinkronizációs pontok az 5.2.2. fejezetben leírtaknak megfelelő formában.

A szenzor mótt a Fourier-együtthatókat 50 mintavételi időnként (T_F) továbbítja a PC felé. Mivel egy mintavételi időköz: $T_s = 4444/8 \cdot 10^6 \approx 0.5555$ msec így az együtthatók küldések gyakorisága: $T_F = 50 \cdot 4444/8 \cdot 10^6 \text{ sec} \approx \frac{1}{36} \text{ sec} \approx 27.8$ msec. A Fourier-együtthatókat tartalmazó csomag a következő adatokat tartalmazza:

1. Csomag azonosítója.
2. A csomag küldésének időpontja, a későbbiekben t_r -vel jelöljük.
3. 5 darab Fourier-együttható értéke a csomag küldésének időpontjában.
4. Az alapharmonikus bázisfüggvény fázisa a Fourier-együtthatók küldésének időpontjában, mely: $\varphi_1(t_r)$.

Ez utóbbi azért szükséges, mert egy periodikus jel esetén önmagában egy fázis információ nem mond semmit, meg kell adni egy referenciát. Amennyiben megadjuk, hogy egy adott időpontban alapharmonikus bázisfüggvény fázisa mekkora, akkor a fázis és a bázisfüggvény elvileg bármely más időpontban ezután kiszámítható amennyiben az f frekvencia állandó:

$$\varphi_1(t) = \varphi_1(t_r) + 2\pi f(t - t_r), \quad (15)$$

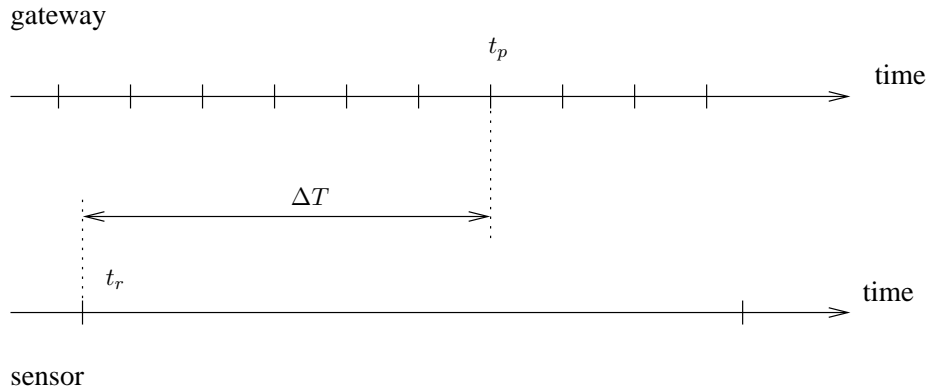
Mivel a frekvencia változhat, ezért szükséges bizonyos gyakorisággal elküldeni a fázist. Az i -edik harmonikus bázisfüggvény fázisa az alapharmonikus fázisának i -szerese:

$$\varphi_i(t) = i\varphi_1(t), \quad (16)$$

a bázisfüggvények pedig:

$$c_i(t) = e^{j\varphi_i(t)}. \quad (17)$$

A mérés során a szenzor és a gateway mótt ugyanazt a jelet mintavételezi (lásd: 2. ábra). A gateway a nyers mintákat továbbítja, míg a szenzor a Fourier-együtthatókat. A Fourier-együtthatók segítségével a megfigyelt jelet visszaállíthatjuk a t_p időpontban a következő lépések



5. ábra. A Fourier-együtthatókból történő jelrekonstrukció időzítési diagramja

végrehajtásával (időadatok magyarázatához lásd az 5. ábrát):

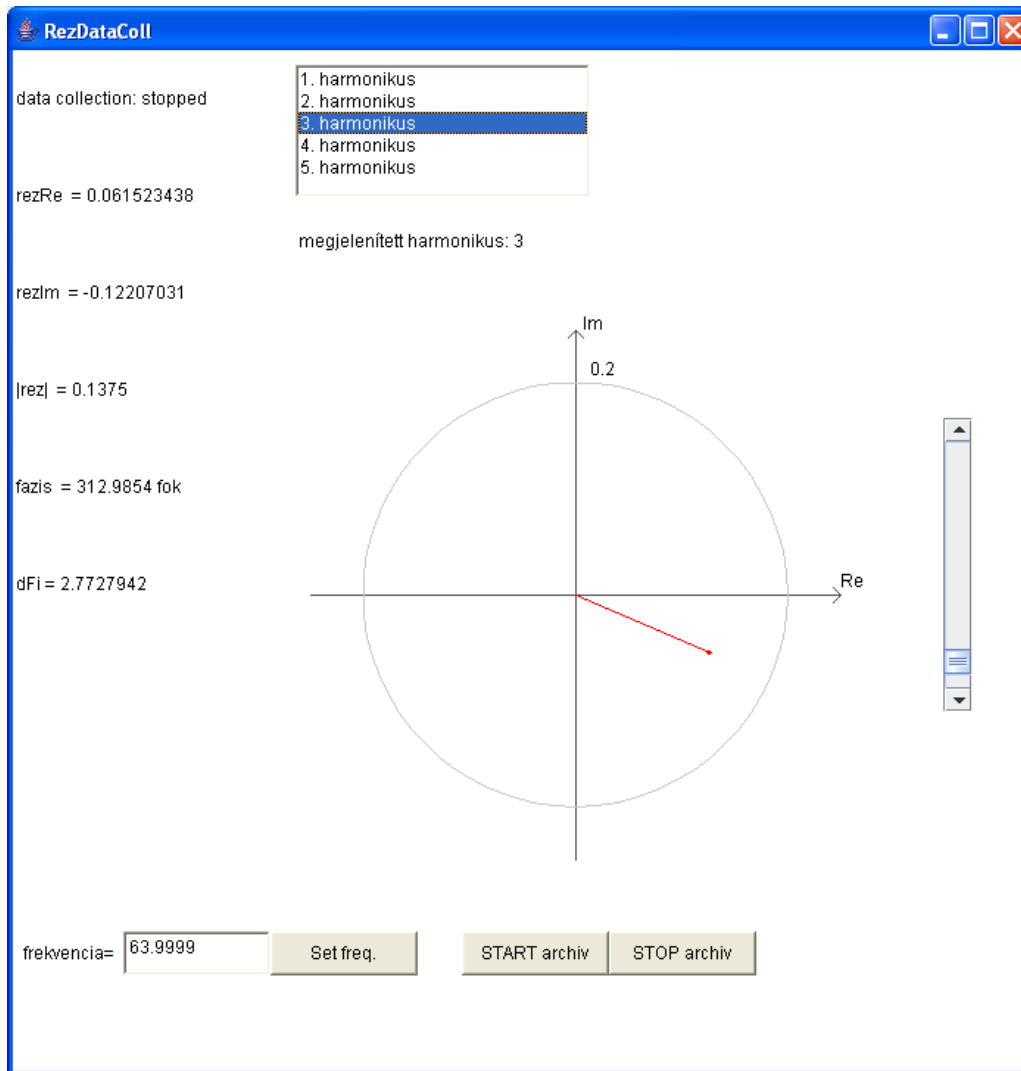
1. Mérjük meg és állítsuk be a jel frekvenciáját, aztán készítsük el a mérési regisztrátumot.
2. Végezzük el a szenzor időbélyegeinek konverzióját.
3. A jel t_p pontban történő előállításához keressük meg a közvetlenül a t_p időpont előtt beérkező legutolsó csomagot. A csomag küldésének időpontja: t_r .
4. Számítsuk ki a csomag érkezése óta eltelt időt: $\Delta T = t_p - t_r$.
5. A csomag küldésekor megadott alapharmonikus fázis értékét ($\varphi_1(t_r)$) módosítsuk a fázis ΔT idő alatt bekövetkező megváltozásával, így megkapjuk az alapharmonikus bázisfüggvény fázisát a t_p időpontban: $\varphi_1(t_p) = \varphi_1(t_r) + 2\pi\Delta T \cdot f$ (lásd: (15)).
6. Számítsuk ki az i -edik harmonikus bázisfüggvény fázisát: $\varphi_i(t_p) = \varphi_1(t_p) \cdot i$ (lásd: (16)).
7. Számítsuk ki a bázisfüggvényeket: $c_i(t_p) = e^{j\varphi_i(t_p)}$ (lásd: (17)).
8. A bázisfüggvények és a szenzor által küldött Fourier-együtthatók (x_i) ismeretében számítsuk ki a jel értékét (12) alapján: $y'(t_p) = 2\Re \left\{ \sum_{i=1}^P c_i(t_p)x_i \right\}$.

Figyelem! MATLABban a \sum művelet hatékonyan megvalósítható vektorszorzással, lásd pl. (6) egyenletet, melyben az egyik vektor transzponáltját (T) kell számítani. Gyakori hiba, hogy MATLABban a $[']$ operátor jelöli a transzponálást, a $[']$ operátor a transzponált konjugáltat jelöli, azonban ez valós jelek esetén nem okoz problémát, csak komplex jelek esetén. Figyeljünk arra is, hogy amennyiben i vagy j változókat ciklusváltozóként használjuk, akkor nem használhatjuk önmagukban a komplex számok képzetes részének jelölésére.

5.3.3. A mérőrendszer felépítése

A rezonátoros megfigyelőt alkalmazó rendszer kezelésére egy PC-s grafikus kezelői felület szolgál, mely a 6. ábrán látható. A kezelői felület a következő lehetőségeket kínálja:

1. Alkalmas a szenzor által számított Fourier-együtthatók megjelenítésére: 6. ábra: pirossal jelölt vektor. A koordináta rendszerben a Re tengely a valós az Im tengely az együtthatók képzetes részét jelölik. A koordináta rendszer melletti csúszka segítségével (jobb oldal) a



6. ábra. A rezonátoros megfigyelőt alkalmazó mérőrendszer kezelői felülete.

tengelyeket skálázhatjuk. A szürke kör szolgál viszonyítási alapként a vektor hosszának becsléséhez.

2. Az ablak tetején lévő listából kiválaszthatjuk azt, hogy melyik Fourier-együtthatót szeretnénk megjeleníteni.
3. A Fourier-együtthatók az ablak bal oldalán numerikusan is láthatóak algebrai és exponenciális formában: **rezRe** és **rezIm** a Fourier-együttható valós és képzetes részét, a **|rez|** és **fazis** a Fourier-együttható hosszát és fázisszögét adja meg.
4. Szintén az ablak bal oldalán látható **dFi** érték, mely megadja, hogy az alapharmonikus Fourier-együttható fázisa mennyit változott az két csomag küldése között. Ebből megállapíthatjuk, hogy mekkora a frekvenciahiba a megfigyelőnek megadott illetve a jel frekvenciája között, értelmezését lásd korábban a frekvenciaméréshez tartozó leírásban.
5. A **frekvencia=** felirat melletti szövegmezőben megadható a megfigyelő frekvenciája. A beírt értéket ENTER-rel vagy a **Set freq.** gomb megnyomásával érvényesítjük. Ekkor a

szenzor móton a LED1 állapota megváltozik.

6. A `START` `archiv` illetve `STOP` `archiv` gombokkal a mótók által küldött adatokat file-ba menthetjük. A `START` `archiv` gomb megnyomásával elkezdjük a mentést a `STOP` `archiv` gombbal pedig leállíthatjuk. Minden indítás után a legutóbbi file-okat felülírja a program, de egy külön könyvtárba minden, a mérés során keletkezett file archiválásra kerül.

A regisztrátumok mentését követően futtassuk a `preprocessResonatorFiles.m` MATLAB scriptet, mely egyszerűen feldolgozható formátumú file-okat generál. A script futtatását követően a következő file-okat használhatjuk.

signalFreq.dat: a kezelői felületen beállított frekvenciát tartalmazza. Ezen file segítségével a feladatok megoldása során az aktuális méréshez tartozó frekvenciát egyszerűen kiolvashatjuk: `sigFreq = load('signalFreq.dat');`;

pre_stmpSync.dat, pre_gwySampFull.dat: az 5.2.2. szakaszban leírtakkal egyező felépítésű és funkciójú file-ok.

pre_sensorResonator.dat: a szenzor által küldött adatsomagokat tartalmazza a következő formában:

```
Msg_ID_1 t1 φ1(t1) ℜ{ $\hat{x}_1(t_1)$ } ... ℜ{ $\hat{x}_5(t_1)$ } ℑ{ $\hat{x}_1(t_1)$ } ... ℑ{ $\hat{x}_5(t_1)$ } rezConf=0
Msg_ID_2 t2 φ1(t2) ℜ{ $\hat{x}_1(t_2)$ } ... ℜ{ $\hat{x}_5(t_2)$ } ℑ{ $\hat{x}_1(t_2)$ } ... ℑ{ $\hat{x}_5(t_2)$ } rezConf=0
.
.
.
```

vagy

```
Msg_ID_1 t1 φ1(t1) ℜ{ $\hat{x}_1(t_1)$ } ... ℜ{ $\hat{x}_9(t_1)$ } ℑ{ $\hat{x}_1(t_1)$ } ... ℑ{ $\hat{x}_9(t_1)$ } rezConf=1
Msg_ID_2 t2 φ1(t2) ℜ{ $\hat{x}_1(t_2)$ } ... ℜ{ $\hat{x}_9(t_2)$ } ℑ{ $\hat{x}_1(t_2)$ } ... ℑ{ $\hat{x}_9(t_2)$ } rezConf=1
.
.
.
```

Tehát az első oszlopban található az adott üzenet azonosítója: `Msg_ID_r`.

A második oszlopban található az üzenet küldésének időpontja: t_r .

A harmadik oszlopban található a szenzoron futtatott megfigyelőben az alapharmonikus bázisfüggvény fázisa az üzenet küldésének időpontjában: $\varphi_1(t_r)$.

Az utolsó (14-edik) oszlopban található `rezConf` flag jelzi, hogy a szenzor az első öt (`rezConf=0`), vagy az első öt *páratlan* Fourier-együtthatót számítja (`rezConf=1`).

A 4...8 -adik oszlopban található a szenzor által számított Fourier-együtthatók valós része ($\Re\{\hat{x}_i(t_r)\}$). Amennyiben a szenzor az első 5 Fourier-együtthatót számítja (`rezConf=0`), akkor ez az $[\Re\{\hat{x}_1(t_r)\}, \Re\{\hat{x}_2(t_r)\}, \Re\{\hat{x}_3(t_r)\}, \Re\{\hat{x}_4(t_r)\}, \Re\{\hat{x}_5(t_r)\}]$ ellenkező esetben (`rezConf=1`): $[\Re\{\hat{x}_1(t_r)\}, \Re\{\hat{x}_3(t_r)\}, \Re\{\hat{x}_5(t_r)\}, \Re\{\hat{x}_7(t_r)\}, \Re\{\hat{x}_9(t_r)\}]$

A 9...13 -adik oszlopban található a szenzor által számított Fourier-együtthatók képzetes része ($\Im\{\hat{x}_i(t_r)\}$). Amennyiben a szenzor az első 5 Fourier-együtthatót számítja

(**rezConf=0**), akkor ez az $[\Im\{\hat{x}_1(t_r)\}, \Im\{\hat{x}_2(t_r)\}, \Im\{\hat{x}_3(t_r)\}, \Im\{\hat{x}_4(t_r)\}, \Im\{\hat{x}_5(t_r)\}]$ ellenkező esetben (**rezConf=1**): $[\Im\{\hat{x}_1(t_r)\}, \Im\{\hat{x}_3(t_r)\}, \Im\{\hat{x}_5(t_r)\}, \Im\{\hat{x}_7(t_r)\}, \Im\{\hat{x}_9(t_r)\}]$
Ezen adatok alapján a Fourier-együtthatók előállíthatóak: $\hat{x}_i(t_r) = \Re\{\hat{x}_i(t_r)\} + j\Im\{\hat{x}_i(t_r)\}$.

5.4. Rezonátor alapú aktív zajcsökkentő (ANC) rendszer tanulmányozása

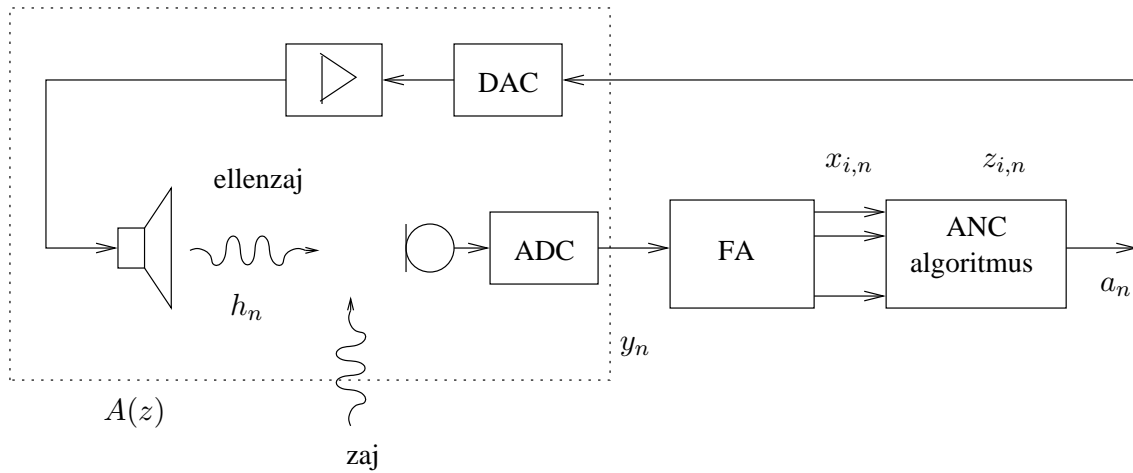
A mérés harmadik részében egy, a rezonátoros megfigyelő adatait felhasználó aktív zajcsökkentő (ANC: Active Noise Control) rendszert ismerünk meg. A zajcsökkentő algoritmus a következő szakaszban kerül bemutatásra.

5.4.1. Rezonátor alapú aktív zajcsökkentő rendszer bemutatása

Az aktív zajcsökkentés alapelve az, hogy egy zajforrás által keltett akusztikus zajt úgy csökkentjük egy adott pontban, hogy a zajjal ellentétes fázisú hangot (ellenzaj) sugárzunk ki. A szuperpozíció elve alapján a zaj és az ellenzaj kioltja egymást. Természetesen valós körülmények között a zajcsökkentés nem lehet teljes mértékű, de körülbelül 10-20 dB értékű zajcsökkentés a gyakorlatban is megvalósítható. Az elnyomandó zajt egy mikrofonnal érzékeljük és az érzékelt jel alapján számíthatóak ki az ellenzaj paraméterei. Esetünkben a zaj érzékelését a vezeték nélküli szenzor látja el. A 7. ábra alapján, ha d_n -el jelöljük a zajt és h_n -el az ellenzaj értékét a mikrofonnál, akkor a maradó zaj (y_n) a következő lesz:

$$y_n = d_n + h_n. \quad (18)$$

A mérés során a zajcsökkentésben az RBO-t futtató szenzor adatait használjuk. Mivel a szenzoron futtatott RBO periodikus jelek feldolgozására alkalmas, így a bemutatásra kerülő algoritmus is periodikus zajok elnyomására használható. Mindebből következik, hogy a kibocsátott ellenzaj is periodikus.



7. ábra. A rezonátoros megfigyelőt alkalmazó aktív zajcsökkentő algoritmus blokkvázlata.

A zajcsökkentő algoritmus blokkvázlata a 7. ábrán látható. Az ábrán látható FA blokk a 4. ábrán látható rezonátor alapú Fourier-analizátort (megfigyelőt) jelöli. A FA Fourier-együtthatókra bontja a jelet, majd egy aktív zajcsökkentő (ANC) algoritmus ez alapján előállítja a megfelelő periodikus ellenzajt.

A 7. ábrán $A(z)$ -vel jelölt, szagatott vonallal határolt rész az akusztikus rendszer átviteli függvénye, mely tartalmaz minden olyan egységet, mely az algoritmus által kiadott jel és a zajt érzékelő szenzor között található, tehát: AD átalakító, erősítő, hangszóró, akusztikus út, mikrofon, jelkondicionáló áramkörök és a rendszerben található szoftveres késleltetések is. A továbbiakban $A(z_i)$ -vel jelöljük az i -edik felharmonikus frekvenciáján az átviteli függvény értékét. Ezen átviteli függvény értékét általában mérésel állapítjuk meg. Az átviteli függvény

mért értéke az i -edik felharmonikus esetén $\hat{A}(z_i)$. Az átviteli függvény az adott rendszerben egyszerűen meghatározható: kiadunk egy ismert fázisú és amplitúdójú jelet, melynek Fourier-együtthatója z_i , és mérjük az erre adott válaszjel Fourier-együtthatóját, mely \hat{x}_i . Az átviteli függvény pedig a válaszjel és a gerjesztés hányadosa: $\hat{A}(z_i) = \frac{\hat{x}_i}{z_i}$. A következőkben használni fogjuk az átviteli függvény inverzét, mely a következő módon definiálható:

$$w_i = \frac{1}{\hat{A}(z_i)} = \frac{1}{\hat{x}_i/z_i} = \frac{z_i}{\hat{x}_i}. \quad (19)$$

Jelöljük a kibocsátott ellenzaj i -edik felharmonikusának Fourier-együtthatóját az n -edik időpontban $z_{i,n}$ -vel, az együtthatókat tartalmazó vektort pedig \mathbf{z}_n -nel. Ekkor az ellenzaj időfüggvénye:

$$a_n = \sum_{i=-P}^P z_{i,n} e^{j\Theta_i n} = \sum_{i=-P}^P z_{i,n} c_{i,n} = \mathbf{c}_{i,n}^T \mathbf{z}_{i,n}. \quad (20)$$

A fenti egyenlet (3) alapján értelmezhető.

Amennyiben ideális működést feltételezünk, akkor az ellenzajt jellemző Fourier-együtthatónak a következő értéket kell felvenni ahhoz, hogy a zajt teljes mértékben kioltssuk. Tegyük fel, hogy a szenzor által mért zaj Fourier-együtthatója x_i , ekkor az ellenzaj Fourier-együtthatója az adott frekvencián:

$$z_{i,n} = -w_i x_i = -\frac{1}{\hat{A}(z_i)} x_i. \quad (21)$$

(21) úgy magyarázható, hogy amennyiben adott egy x_i Fourier-együtthatójú jel (az elnyomandó zaj), akkor az egy vele ellentétes fázisú, tehát $-x_i = x_i e^{j180^\circ}$ jellel oltható ki. Mivel az akusztikus rendszer $A(z_i)$ -szeresére módosítja a kiadott jelet, így ahhoz, hogy $-x_i$ jelenjen meg a mikrofonnál, a kiadott jel Fourier-együtthatója $z_{i,n} = -\frac{1}{\hat{A}(z_i)} x_i = -w_i x_i$ kell, hogy legyen.

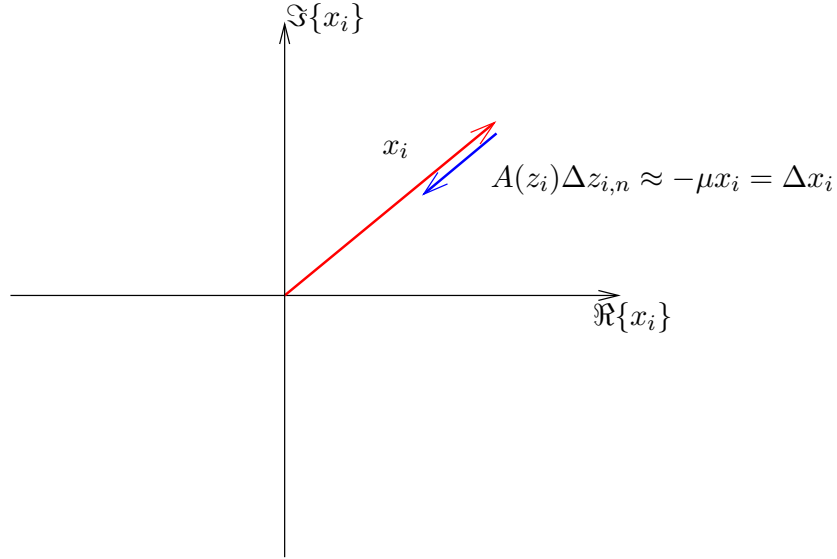
Ha például a zaj alapharmonikus Fourier-együtthatója: $x_1 = 2e^{j45^\circ}$, és az akusztikus rendszer a kiadott jelet $A(z_1) = 0.2e^{j20^\circ}$ -kal módosítja, ebben az esetben tehát a kiadott ellenzaj alapharmonikus Fourier-együtthatóját $z_{1,n} = -\frac{1}{A(z_1)} x_1 = -\frac{1}{0.2e^{j20^\circ}} 2e^{j45^\circ} = -10e^{j25^\circ}$ -re kell állítani. Az ellenzaj Fourier-együtthatóinak ily módon történő beállítása gyakorlatban közvetlenül nem alkalmazható két ok miatt: egyrészt a zaj paraméterei változhatnak, így a kiszámított ellenzaj nem marad optimális, másrészt az átviteli függvény mérése nem tökéletes (pl. mérési zaj miatt), így a kiszámított ellenzaj nem optimális a valóságban.

A zajcsökkentő algoritmus az ellenzaj Fourier-együtthatóinak optimális értékét egy iteratív algoritmussal közelíti meg. A jelfeldolgozó algoritmus a kimeneti Fourier-együtthatókat a következő módon frissíti minden időpontban:

$$z_{i,n+1} = z_{i,n} - \mu w_i x_i; \quad i = [1 \dots P], \quad (22)$$

ahol μ egy bátorsági tényező, melyre: $0 < \mu < 1$. Látható, hogy az algoritmus minden alkalommal olyan irányba módosítja $z_{i,n}$ -t, mely csökkenti a zajt. Az algoritmus szemléletesen úgy értelmezhető, hogy ha megfontoljuk, hogy amennyiben a beavatkozó jel Fourier-együtthatóját $\Delta z_{i,n} = -\mu w_i x_i$ -al módosítjuk, akkor az az akusztikus rendszeren áthaladva $A(z_i)$ -szeresére módosul. Összességében tehát az x_i Fourier-együtthatóval rendelkező érzékelt zaj $\Delta x_i = A(z_i) \Delta z_{i,n} = -\mu A(z_i) \frac{1}{A(z_i)} x_i \approx \mu x_i$ -vel módosul, tehát μ -ad részével csökken, ahogyan a 8. ábrán látható.

(22) felfogható úgy is, mint egy komplex értékekkel dolgozó integráló szabályozó.



8. ábra. ANC algoritmus magyarázata. x_i (piros nyíl): a zaj Fourier-együtthatója. Δx_i (kék nyíl): a zajcsökkentő algoritmus egy iterációja utáni csökkenés a zaj Fourier-együtthatójában.

Meg kell jegyeznünk, hogy az előbbi magyarázat $\mu \ll 1$ esetén érvényes, ugyanis a felhasznált összefüggések állandósult állapotban érvényesek, mely akkor indokolt, ha a változások lassúak, tehát $\mu \ll 1$. Amennyiben az algoritmust manuálisan valósítjuk meg a PC-s kezelői felületet használva, akkor $\mu = 1$ is használható, hiszen ekkor a kimenő jelet leíró $z_{i,n}$ Fourier-együtthatókat elég ritkán tudjuk frissíteni ahhoz, hogy beálljon az állandósult állapot. $\mu = 1$ esetén (22)-et végrehajtva ideális esetben minden végrehajtás után kioljuk a zajt.

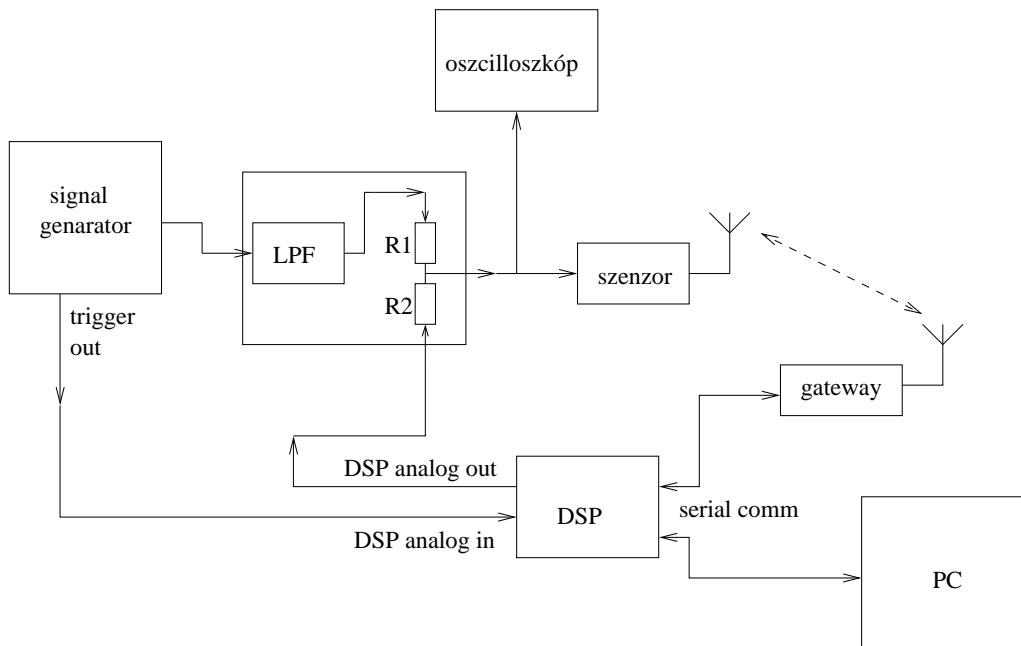
(22) algoritmus helyes működéséhez szükséges, hogy az átviteli függvényt helyesen mérjük, tehát: $\hat{A}(z_i) \approx A(z_i)$. Belátható, hogy az algoritmus stabil marad, ha az átviteli függvényt legalább $\pm 90^\circ$ pontossággal ismerjük.

5.4.2. A mérőrendszer felépítése

A zajcsökkentő rendszerhez tartozó mérés során a 9. és a 10. ábrán látható elrendezéseket alkalmazzuk. Alapvetően a két rendszer működése a következő. Zajként egy függvénygenerátor által előállított jelet alkalmazunk. A szenzor érzékeli az elnyomandó zajt és előállítja a Fourier-együtthatókat, majd a rádió segítségével periodikusan elküldi a gateway-en keresztül a DSP felé. A DSP 2 kHz-es jelfeldolgozási sebességgel üzemel, és a (22) egyenlet alapján futtatja a zajcsökkentő algoritmust. A DSP a szenzor adatait továbbítja soros porton keresztül a PC felé, így megjeleníthetők az eredmények.

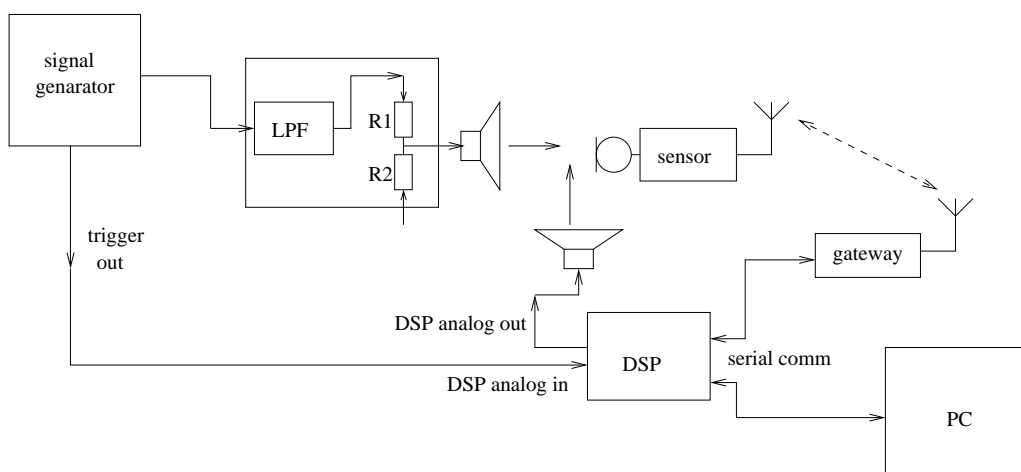
A 9. ábrán látható esetben a kiadott jel és a zavarjel összegzése nem akusztikailag valósul meg, hanem két ellenállás segítségével egy egyszerű összegző kapcsolást használva. Ez az elrendezés lehetővé teszi az algoritmus tulajdonságainak megismerését egyszerű körülmények között, ugyanis az algoritmus „próbálgatása” akusztikus jelekkel körülményesebb lenne. Ez a kapcsolat arra is példát mutat, hogy hogyan lehet például elektronikus zavarokat kioltani az algoritmus segítségével (pl. a hálózati 50 Hz-es jel által okozott zavarjelek).

A 10. ábrán látható elrendezésben a kialakítás megfelel a zajcsökkentő rendszerek általános felépítésének, és egyszerűen kialakítható a 9. ábrán látható összeállításból. Ebben a rendszerben már tesztelhetjük az algoritmust az akusztikus zajcsökkentés esetén is.



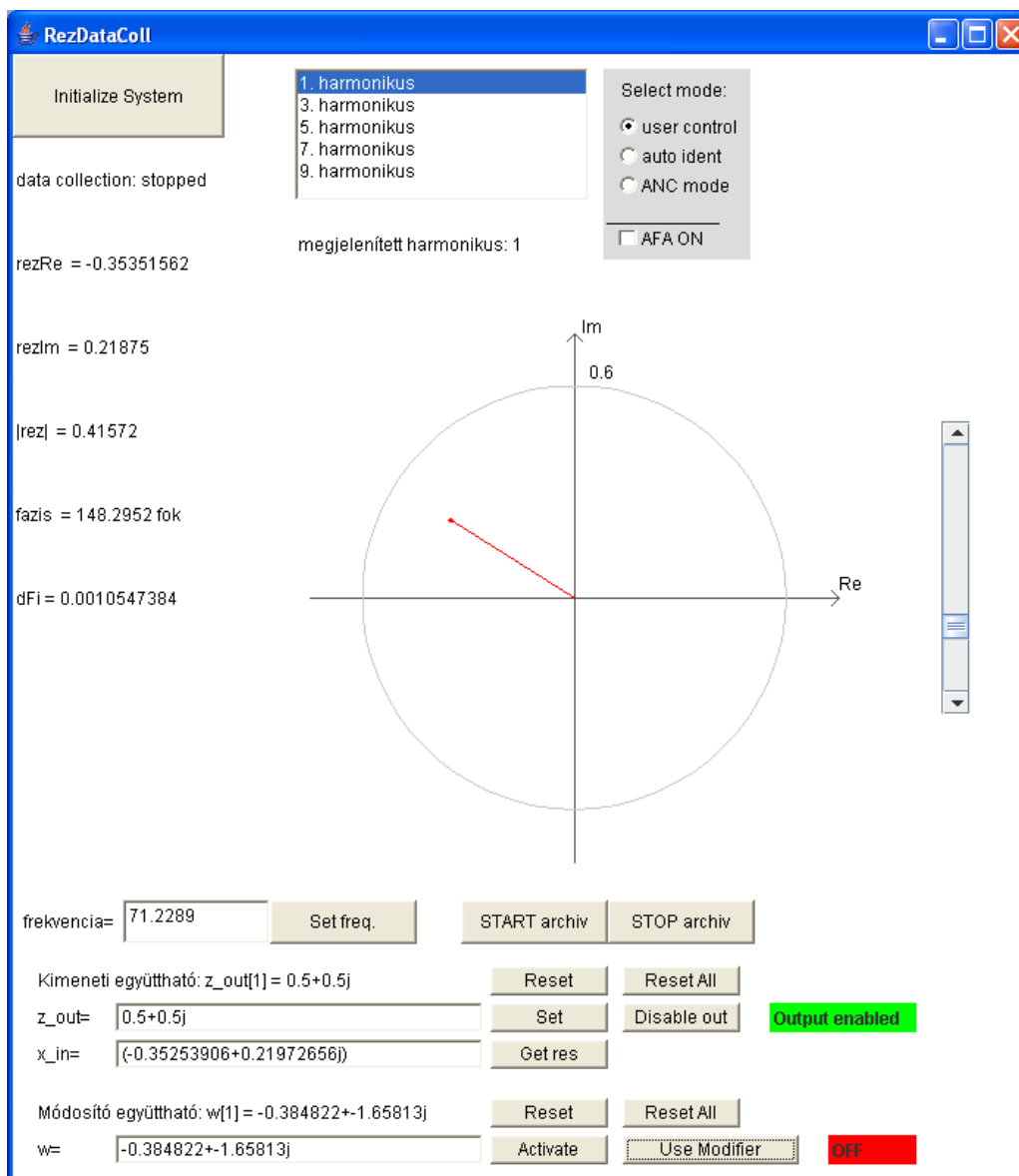
9. ábra. A rezonátoros megfigyelőt alkalmazó aktív zajcsökkentő rendszer fizikai konfigurációja (algoritmus tesztelés).

Megfigyelhetjük, hogy a 9. és 10. ábrán a DSP bemenetére kötjük a jelgenerátor trigger kimenetét. Ez a zajként alkalmazott jel frekvenciájának méréséhez használható. A méréshez a már bemutatott AFA (Adaptive Fourier Analyzer) algoritmust használja. Ez amiatt szükséges, mert manuális üzemmód esetén az emberi reakcióidő bizonyos esetekben nem teszi lehetővé több paraméter megfelelő gyorsasággal történő beállítását.



10. ábra. A rezonátoros megfigyelőt alkalmazó aktív zajcsökkentő rendszer fizikai konfigurációja.

A zajcsökkentő rendszer kezeléséhez a rezonátoros adatgyűjtő rendszer esetén bemutatotthoz hasonló PC-s kezelői felület használható (lásd 11. ábra). Az adatgyűjtő rendszerhez képest a következő extra funkciók állnak rendelkezésre:



11. ábra. A rezonátoros megfigyelőt alkalmazó aktív zajscsökkentő rendszer kezelői felülete.

1. Üzem mód beállítás: a **Select mode** felírat alatt találhatóak a főbb üzemmódok, melyek röviden a következők.

user control: minden paraméter manuálisan állítható.

auto ident: a DSP automatikusan elkezd az átviteli függvény mérését. A mérés léptetett szinuszzel történik, és a DSP belső memóriájában tárolja a mért értékeket. Miután minden frekvencián végrehajtotta a mérést a DSP, az **ANC mode**-ba történő váltással indul a zajscsökkentés. Ahhoz, hogy a DSP által mért paramétereket használjuk, kapcsoljuk ki a **w** módosító együtthatókat (**Don't use Modifiers**), ellenkező esetben az általunk manuálisan mért w_i -ket használja a rendszer.

ANC mode: a DSP-n bekapcsoljuk az aktív zajscsökkentő algoritmust.

AFA ON: az Adaptív Fourier Analizátor algoritmust kapcsolhatjuk be a DSP-n, mely au-

tomatikusan méri a jel frekvenciáját a jelgenerátorról a DSP-re vezetett triggerkimenet segítségével. Figyelem: az AFA a PLL-ekhez hasonlóan rendelkezik bizonyos befogási tartománnyal. Ez körülbelül 20 Hz és 200 Hz között található, de függhet például attól is, hogy milyen frekvenciáról mekkorára változik a jel frekvenciája. Amennyiben nem találja a DSP a frekvenciát, akkor próbálkozzunk annak megváltoztatásával, illetve az AFA mód ki- és bekapcsolásával. Az AFA által mért frekvencia bizonyos gyakorisággal megjelenik a frekvenciabállítást lehetővé tévő szövegdobozban.

2. Beállíthatóak a beavatkozó jel (z_i) Fourier-együtthatói a `z_out=` felirattal jelzett szövegdobozban. Mindig ahhoz a harmonikushoz tartozó együtthatót szerkesztjük, melyet éppen megjelenítettünk. A beírt értéket `ENTER`-rel vagy a `Set` gomb megnyomásával érvényesítjük. Ez felhasználható identifikáció során és zajcsökkentéshez. Identifikációnál például ha kiadunk egy $0.8 + 0j$ együtthatóval rendelkező jelet és $0.5 + 0.77j$ a mért jel, akkor az átviteli függvény értéke: $(0.5 + 0.77j)/(0.8 + 0j) = 0.625 + 0.9625j$.

A szövegmezőben megadhatóak összeadás és kivonás műveletek illetve zárójelezés. Ennek segítségével manuálisan is egyszerűen megvalósítható (22) algoritmus. Az éppen aktuálisan mért jel Fourier-együtthatói a `Get res` gomb segítségével szövegesen szerkeszthető formában kerülnek be a gomb melletti szövegdobozba.

A kimenet letiltható a `Disable out` gomb segítségével; ANC módban ez a zajcsökkentő algoritmus leállítását is jelenti. Az összes, illetve az aktuálisan kijelzett kimeneti Fourier-együttható nullázható a `Reset All` illetve a `Reset` gombok segítségével. A `Get res` gomb megnyomásával a gomb melletti szövegdobozba kerül a szenzor által utoljára küldött Fourier-együttható értéke.

3. Módosító együttható (w) jelentése a következő: amennyiben engedélyezzük a módosító együtthatók használatát, a szenzor által mért eredményeket a DSP megszorozza szenzor által mért együtthatókat az itt megadott értékekkel. Amennyiben a módosító együtthatók helyére beírjuk az adott frekvencián az átviteli függvény inverzét, tehát w_i -t (lásd (19) egyenlet), abban az esetben kompenzáljuk az átviteli függvény hatását, tehát olyan mintha a DSP kimenete közvetlenül a szenzor bemenetére kerülne. Ez megkönnyíti a zajcsökkentés manuális implementálását, ugyanis nem kell a (22) egyenletben található szorzást manuálisan elvégezni, ezt megteszi helyettünk a DSP.

Az együtthatók reset értéke (`Reset/Reset All`) egy.

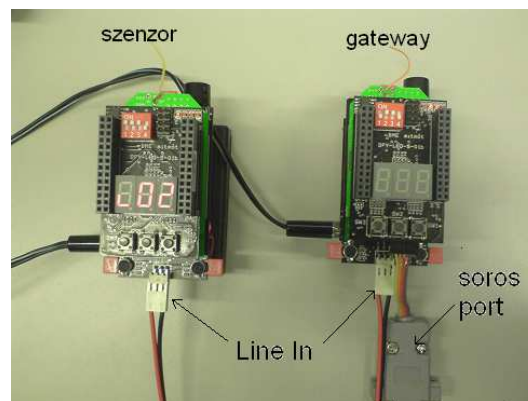
Amennyiben `ANC mode`-ban vagyunk, a DSP számára szükségesek a w_i együtthatók (az átviteli függvény inverzei az adott frekvenciákon) (22) számításához. Amennyiben a módosító együtthatók (w) használata engedélyezett, akkor a manuálisan beírt értékeket használja a DSP, egyébként az általa mért értékeket. Értelemszerűen ekkor a zajcsökkentő algoritmus indítása előtt egy automatikus identifikációt kell végezni.

4. A `START archiv` illetve `STOP archiv` gombokkal a DSP által küldött adatokat file-ba menthetjük. Ebben az esetben a gateway módt adatai nem kerülnek mentésre, mert azt a DSP nem továbbítja a PC felé.
5. Az `Initialize system` gomb (11. ábra bal felső sarok) a rendszer inicializálására szolgál. Akkor érdemes használni, amikor a rendszer valamelyik részét újraindítjuk, vagy rendellenes működést tapasztalunk, ekkor ugyanis a PC alaphelyzetbe állít minden funkciót, és a DSP felé is elküldi a megfelelő beállításokat.

5.5. Felhasznált eszközök összefoglalása

Ebben a fejezetben összegezzük a mérés során felhasznált eszközök bemutatását.

Mitmótok. A rendszerben két darab mitmótot alkalmazunk; egyik szenzorként funkcionál a másik pedig bázisállomásként és érzékelőként is funkcionál. Mindkét mót rendelkezik egy IO kártyával (DPY-LED-S-01b), egy akusztikus szenzorkártyával és egy 2.4 GHz-es ZigBee rádiós kártyával. Az akusztikus szenzorkártya képes akusztikus jelek érzékelésére valamint található rajta közvetlen, erősítés nélküli vonalbemenet, melyre tetszőleges analóg jelet vezethetünk $0 V_{pp} - 3.3 V_{pp}$ tartományban. A kártya AC csatolású, körülbelül 20 Hz-nél található az alsó törésponti frekvencia.



12. ábra. A mérésben használt mitmótok felépítése

A mótokon található kapcsolók és nyomógombok segítségével konfigurálhatóak működés közben. A DIP kapcsolón található 4-es kapcsoló segítségével választhatunk, hogy a mótok a mikrofon vagy a line-in bemenetet mintavételezzék. A mintavételezés frekvenciája $f_s = 8 \cdot 10^6 / 4444 \approx 1800.2$ Hz.

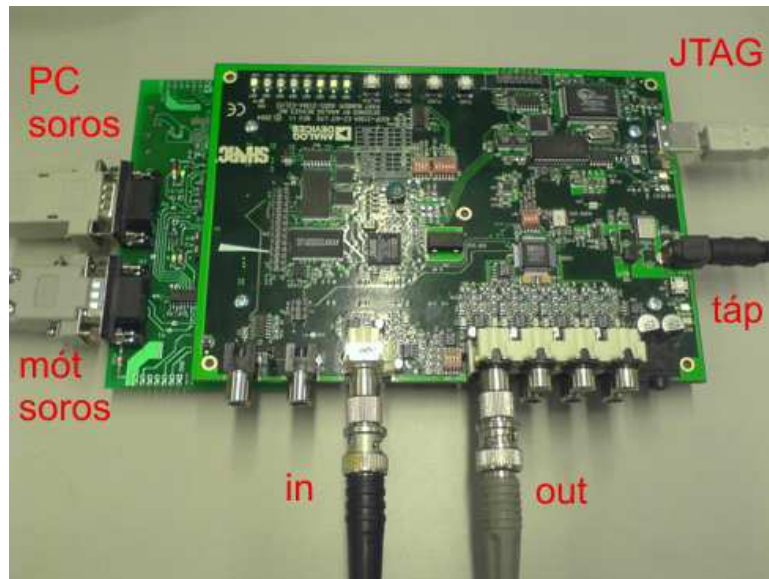
A szenzor mót kétféle üzemmódban képes üzemelni. Az üzemmódot a mót bekapcsolásakor (vagy reset-et követően) lehet beállítani az SW1 illetve SW2 gombokkal lépkedve az üzemmódok között. Az SW3 gombbal nyugtázhatjuk az üzemmódot. 1-es üzemmódban a szenzor csupán a mintákat továbbítja a gateway felé, 2-es üzemmódban a Fourier-együtthatókat számítja.

Amennyiben a szenzor mót a Fourier-együtthatókat számítja, akkor az 1-es kapcsoló segítségével választhatjuk ki, hogy az [1...5] harmonikusokat számítsa, vagy az [1, 3, 5, 7, 9] harmonikusokat.

A mótokon található RESET gomb segítségével újraindíthatóak a mótok. A RESET gomb akkor használandó, ha új üzemmódban szeretnénk indítani a rendszert vagy rendellenes működést tapasztalunk.

A szenzor móton a LED1 állapotváltozása jelzi egy új frekvencia beállítását. A frekvencia a 11. és a 6. ábrán látható kezelői felületek segítségével állítható be.

DSP. A mérés során használt DSP egy Analog Devices által forgalmazott, ADSP-21364-es típusú, 32 bites, lebegőpontos jelfeldolgozó processzor. A processzor egy fejlesztői kártyán található, melyen találhatóak a mérés szempontjából fontos analóg-digitális (ADC) és digitális-analóg átalakítók (DAC). A kártyán található egy beépített JTAG interface, mely segítségével a



13. ábra. A mérésben használt DSP kártya

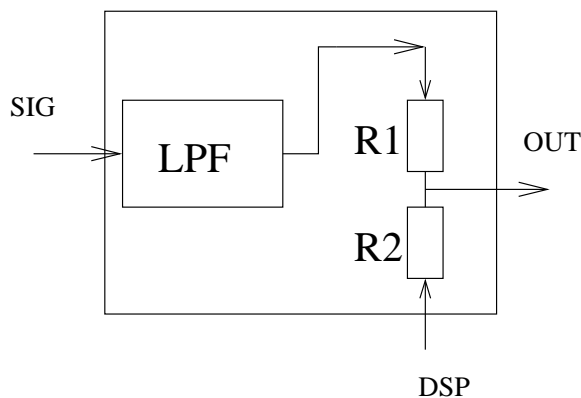
PC-ről programozhatjuk a DSP-t és debuggolhatjuk a futó programot (a DSP-t USB segítségével kapcsolhatjuk a PC-hez programozás céljából). A DSP-t a VisualDSP fejlesztői környezetben keresztül kezelhetjük. A fejlesztői környezetről egy rövid leírás található a tárgy honlapján.

A processzor a mérés során 2 kHz-es mintavételi frekvenciával üzemel.

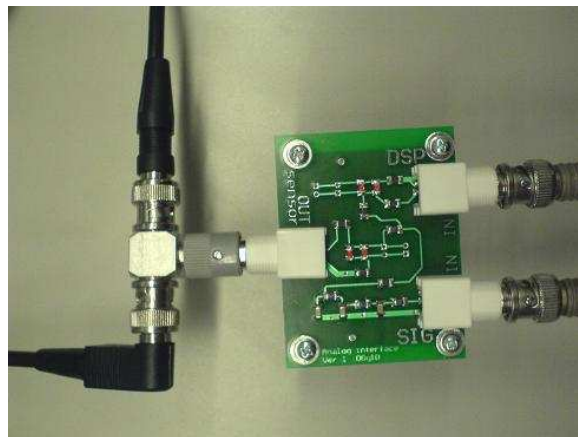
A DSP-t használó mérésekhez szükséges szoftverek rendelkezésre állnak a PC-n.

Mérés során a DSP működését nagyrészt egy, a 11. ábrán látható program segítségével vezérelhető soros port segítségével. Amennyiben új frekvenciát állítunk be, annak sikerességét a LED1 jelzi. Amennyiben más jellegű parancsot küldünk, azt a LED2 jelzi. A mótokkal való kommunikáció a LED7 és LED8 segítségével ellenőrizhető.

Analóg interface kártya. Az analóg interface kártya (14. ábra) az egyes egységek tehát a jelgenerátor, a DSP és a mótok közötti összeköttetést teszi lehetővé az analóg jelek számára. A jelgenerátor felől tartalmaz egy egyszerű RC aluláteresztő szűrőt (LPF) a jel sávkorlátozásához.



(a) Analóg interface kártya blokkvázlata



(b) Analóg interface kártya fizikai kialakítása

14. ábra. Analóg interface kártya

5.6. Mérési feladatok

A mérési elrendezések összeállításakor ügyeljünk arra, hogy először tápfeszültséget csatlakoztassuk az eszközökhöz, és csak aztán kössük be a jel illetve adatvezetéseket. Minden esetben ellenőrizzük a mótokon a kapcsolók állását, mert azok alapvetően befolyásolják a működést.

1. Mintavételező rendszer vizsgálata

- 1.1. Állítsa össze a 2. ábrán látható rendszert! A szenzor mótot 1-es üzemmódban kell elindítani. A szenzoron a DIP kapcsolók állása 1...4 -ig a következő: [ON ON ON OFF]. A gateway-en a DIP kapcsolók állása 1...4 -ig a következő: [ON ON ON OFF]. A függvénygenerátoron érdemes először 40 Hz és 100 Hz közötti jelet beállítani, mely elősegíti a jel vizuális kiértékelését.
- 1.2. Készítsen mérési regisztrátumot! A regisztrátum hossza az első próbák esetén ne legyen néhány sec-nál hosszabb, mert a nagy mennyiségű adat feldolgozása időigényes lehet. A mérés befejeztét követően futtassuk le a `preprocessSampleFiles.m` MATLAB scriptet, mely egyszerűen feldolgozható formátumúvá konvertálja mentett adatokat. Továbbiakban a 'pre_' előtagú file-okat kell feldolgozni: `pre_stmpSync.dat`, `pre_gwySampFull.dat`, `pre_sensorDataFull.dat` (lásd: a 7. oldalon)
- 1.3. Vizsgálja meg a szinkronizációs pontokat a `pre_stmpSync.dat` file alapján, és ábrázolja a két mót lokális idejét egymáshoz képest! Határozza meg az időbélyeg-transzformációhoz szükséges paramétereket (lásd: (2)), majd végezze el az időbélyeg-transzformációt a meghatározott paraméterek alapján az 5. oldalon leírt módszer felhasználásával! Ábrázolja a regisztrátumok időfüggvényeit az időbélyeg-transzformáció előtt és után, és értékelje az eredményeket! Készítsen mérési regisztrátumot miközben folyamatosan változtatja a jel amplitúdóját!

2. Rezonátoros adatgyűjtő rendszer vizsgálata

- 2.1. A mérési összeállítást változatlanul hagyva resetelje a mótokat! A szenzor mótot 2-es üzemmódban kell elindítani. Ebben az esetben a szenzor a Fourier-együtthatókat továbbítja a PC felé. A szenzoron a DIP kapcsolók állása 1...4 -ig a következő: [ON ON ON OFF]. A gateway-en a DIP kapcsolók állása 1...4 -ig a következő: [ON ON ON OFF]. A függvénygenerátoron érdemes először 40 Hz és 100 Hz

közötti jelet beállítani, mely elősegíti a jel vizuális kiértékelését. Az adatgyűjtést befejezve a `preprocessResonatorFiles.m` script meghívásával generálhatjuk az egyszerűen feldolgozható formátumúvá konvertált file-okat: `pre_stmpSync.dat`, `pre_gwySampFull.dat`, `pre_sensorResonator.dat`, `signalFreq.dat` (lásd: a 15. oldalon)

2.2. A PC-s kezelői felületen állítson be egy becsült frekvenciát (pl. oszcilloszkópon mérve a periódusidőt)! Mérje meg pontosan a jel frekvenciáját az alapharmonikushoz tartozó Fourier-együttható segítségével a következő módon:

- a) Az alapharmonikus Fourier-együttható forgási irányát figyelve növelje illetve csökkentse a frekvenciát, míg a Fourier-együtthatót reprezentáló vektor forgása megáll.
- b) *Kiegészítő feladat* Az alapharmonikus Fourier-együttható forgásának periodusidejét mérve (manuálisan). (Az idő mérésére használhatja pl. a MATLAB `tic` és `toc` függvényeit).
- c) *Kiegészítő feladat* Az alapharmonikus Fourier-együttható fázisát (lásd: `anngle` és `unwrap` függvények) ábrázolja és illesszen egyenest a fázis időfüggvényére. (14)-t felhasználva az egyenes meredeksége a körfrekvencia-eltéréssel egyezik meg.

2.3. Készítsen mérési regisztrátumot! A regisztrátum hossza az első próbák esetén ne legyen néhány sec-nál hosszabb, mert a nagymennyiségű adat feldolgozása időigényes lehet. A gateway adataiból távolítsa el a DC komponenst (vonja ki az átlagot), mert az csupán az unipoláris ADC miatt van a jelben, a rezonátoros megfigyelő a DC komponenst viszont nem számítja. Az előzőekben leírtak alapján végezze el az időbélyegtranszformációt és állítsa elő a megfigyelt jelet a Fourier-együtthatókból a gateway mintavételi időpontjaiban! A rekonstrukció algoritmus a 13. oldalon olvasható. A rekonstrukcióhoz szükséges a jel frekvenciájának ismerete, ezt a `signalFreq.dat` fileból olvashatjuk ki. Hasonlítsa össze a mintavételezett és a Fourier-együtthatókból rekonstruált jelet! Hasonlítsa össze a visszaállított és mintavételezett jelet különböző frekvenciájú és változó amplitúdójú jelek esetén is! Hogyan tudja a rendszer követni az amplitúdóváltozást?

2.4. *Kiegészítő feladat:* Vizsgálja meg, hogy különböző frekvenciabecslési hibáknál milyen pontosan lehet visszaállítani az időtartománybeli jelet! (Frekvenciabecslési hiba alatt azt értjük, hogy mennyivel tér el a jel valódi frekvenciája a beállított értéktől.)

2.5. *Kiegészítő feladat:* Vizsgáljuk meg, hogy hogyan változik a becsült amplitúdó, amennyiben a valóditól eltérő frekvenciát állítunk be, pl. ± 0.01 Hz, ± 0.1 Hz, ± 0.5 Hz, ± 1.0 Hz, ± 2.0 Hz, ± 5.0 Hz, ± 10.0 Hz ...

2.6. *Kiegészítő feladat:* Vizsgálja meg, hogy milyen hibával állítható vissza egy mért jel, amennyiben a szenzor az első öt, illetve az első öt *páratlan* Fourier-együtthatót méri! Vizsgálójelként használjon szinusz, háromszög illetve négyszög jelet! Azt, hogy a szenzor mót melyik együtthatókat számítsa a szenzor móton található 4-es kapcsolóval lehetséges kiválasztani (ON: 1...5, OFF: 1,3,5,7,9). Ezt a mentett file utolsó oszlopa is jelzi, ahogyan az bemutatásra került a 15. oldalon.

3. Rezonátor alapú aktív zajcsökkentő rendszer vizsgálata

3.1. Figyelem! A jelvezetékeket érdemes a tápfeszültségek bekapcsolása után csatlakoztatni a kártyákhoz. Állítsa össze a 9. ábrán látható rendszert! (A fizikai realizáció

a 13. ábra alapján állítható össze.) Ebben az esetben a szenzor mót a Fourier-együtthetők számítja, tehát 2-es üzemmódban kell üzemeltetni. A szenzoron a DIP kapcsolók állása 1...4 -ig a következő: [ON ON ON OFF]. A gateway-en a DIP kapcsolók állása 1...4 -ig a következő: [ON ON ON OFF]. A DSP a VisualDSP fejlesztői környezet segítségével programozható. Fontos: a mérés során a VisualDSP 4.5-ös verziót használjuk kompatibilitási okok miatt. A VisualDSP fejlesztői környezetről rövid összefoglalót találhat a tárgy honlapján. A fejlesztői környezet indításakor zárjon be *minden projektet* és *minden ablakot*, és az adott méréshez tartozó projektet nyissa csak meg, így elkerülhető, hogy régi, vagy átalakított file-okat használjunk. A betöltendő projekt neve: Lab_IF_ANC.

A mérés teljes tartama alatt a függvénygenerátor maradjon az analóg összegző kártyához csatlakoztatva! Amennyiben nincsen szükség külső jelre akkor tiltsuk le a kimenetet (esetleg vegyük a kimenet amplitúdóját a lehető legkisebbre). Ez azért fontos, mert a függvénygenerátor kimenő ellenállásának jelenléte illetve eltávolítása megváltoztatja a rendszer átvitelét. Ugyanez vonatkozik arra, ha más eszközt csatlakoztatunk a rendszerhez (pl. PC hangkártya).

- 3.2. Próbálja ki a rendszert, vizsgálja meg, hogy hogyan állítható elő DSP segítségével egy tetszőleges jel egy adott frekvencián! Milyen különbség van a kiadott és a szenzor által mért jel Fourier-együtthetői között? Próbáljon előállítani olyan jelet, hogy a szenzor által érzékelt jel például $0.5 + 0.2j$ legyen! Nézze meg, hogy a módosító együtthetők használata hogyan egyszerűsíti a jelek előállítását!
- 3.3. Minden felharmonikus frekvencián mérje meg a visszacsatolás átviteli függvényét! (Állítson be egy tetszőleges gerjesztő jelet és mérje meg a választ.) Gerjesztőjelként érdemes minél nagyobb amplitúdójú jelet beállítani. Az amplitúdó ne érje el az egységnyit, mert az már telítést okoz.
- 3.4. Valósítsa meg manuálisan (a kimeneti Fourier-együtthetők kézzel megadva) a (22) egyenlet szerinti zajcsökkentő algoritmust $\mu = 1$ paraméterrel! Vizsgálja meg, hogy milyen nehéz megvalósítani az algoritmust az automatikus AFA használatával és anélkül! Vizsgálja meg, hogy milyen gyorsan nő meg újra az elnyomott zaj AFA módban és anélkül, amennyiben nem változtatjuk a beavatkozó jelet? Kapcsolja ki a zajforrást és vizsgálja meg a kiadott ellenzajt! Teljesül-e, hogy ellentétes fázisúak?
- 3.5. Implementálja DSP-n a (22) egyenletben megadott zajcsökkentő algoritmust! Az implementáció a VisualDSP 4.5 fejlesztői környezetben C nyelven lehetséges. A fejlesztői környezetben nyissa meg a ANCResonatorUpdate.c filet (a baloldali Project ablak, Sources könyvtár), az ebben található ANCResonatorUpdate függvény a DSP minden mintavételi időpontjában meghívódik ANC módban, így itt megvalósítható a zajcsökkentő algoritmus. A függvényben deklarálhatók saját változók (pl. ciklus implementálásához), valamint a következő változókat használhatjuk a (22) képletben található változók elérésére:

```

 $\Re\{z_i\}$ : o_rez_x_re_ptr[i],  $\Im\{z_i\}$ : o_rez_x_im_ptr[i];
 $\Re\{w_i\}$ : modifiers_re[i],  $\Im\{w_i\}$ : modifiers_im[i];
 $\Re\{x_i\}$ : rez_x_re_orig_ptr[i],  $\Im\{x_i\}$ : rez_x_im_orig_ptr[i];
 $\mu$ : nu0.

```

Ne feledjük, hogy (22) képletben μ kivételével minden szám komplex, így a műveleteket el kell végezni külön a valós és képzetes részre felhasználva a komplex számok szorzásának szabályát: $(a + jb)(c + jd) = (ac - bd) + j(ad + bc)$.

Fontos! Az algoritmus helyes működéséhez szükséges a w együtthatók megfelelő beállítása! Állítsa be a PC-s kezelői felületen w együtthatókat úgy, hogy kompenzálják az átviteli függvény hatását, és engedélyezze a használatukat!

- 3.6. Vizsgálja meg a zajcsökkentő rendszer működését, pl. beállási idő! Ábrázolja a mért Fourier-együtthatók trajektóriáját az időben! (Mivel komplex számok időfüggvényét kell ábrázolni, így a `plot3` nevű MATLAB függvény használata javasolt, úgy, hogy az első paraméter az idő, a második az adott változó valós a harmadik paraméter az adott változó képzetes része.)
- 3.7. Változtassa meg a w együtthatókat (w_i), úgy hogy az eredeti szögtől 1° , 10° , 20° ... eltérés legyen! (AFA ON beállítást érdemes használni, hogy a frekvenciahiba hatását kiküszöböljük.) A w_i együtthók szögét úgy módosíthatjuk egy $\Delta\phi$ szöggel, ha megszorozzuk $e^{j\Delta\phi}$ -vel. Mit tapasztal a beállítás során? Milyen úton jut el a kioltott zaj Fourier-együtthatója a nulla pontba? Ábrázoljon ebben az esetben is néhány jellegzetes trajektóriát!
- 3.8. Vizsgálja meg, hogy hogyan változik az átviteli függvény a frekvenciával! Milyen frekvenciaváltozás esetén érjük el a kritikus értéket a fázishibában az egyes harmonikusok esetén, tehát mikor lesz instabil a rendszer?
- 3.9. Vizsgálja meg az akusztikus zajcsökkentő rendszert! A mérési összeállítás a 10. ábrán látható. A szenzoron a DIP kapcsolók állása 1...4 -ig a következő: [ON ON ON ON], így a szenzor a mikrofon jelét érzékeli. Mérje meg a zajcsökkentés mértékét egy külső mikrofonnal!