

# Tartalomjegyzék

Tartalomjegyzék.....	1
1. Absztrakt .....	3
1.1. Abstract .....	5
2. Bevezetés.....	7
3. Zajcsökkentő eljárások LMS algoritmussal .....	11
3.1. A Wiener-szűrők és az LMS algoritmus .....	11
3.1.1. Az akusztikus átvitel mérése .....	15
3.2. A Filtered-X LMS algoritmus [1] .....	20
3.2.1. A Filtered-X LMS algoritmus analízise .....	21
3.3. A Filtered- $\epsilon$ LMS algoritmus [1].....	24
3.3.1. Analízis.....	27
3.4. A Filtered-EX LMS algoritmus [6].....	29
3.4.1. Analízis.....	30
4. Periodikus jelmodell alapú eljárások.....	33
4.1. Jelmodell alapú megfigyelő, a rezonátoros struktúra.....	35
4.2. Adaptív Fourier-analízis.....	39
4.3. Rezonátoros zajcsökkentő eljárások elemzése .....	40
4.3.1. Stabilitás .....	42
4.3.2. Konvergenciasebesség növelése .....	44
4.3.3. Az akusztikus átvitel mérése [11] .....	45
5. On-line identifikáció .....	47
5.1. On-line identifikáció Filtered-X, Filtered- $\epsilon$ és Filtered-EX LMS rendszerekben....	47
5.2. On-line identifikáció Unified-error Filtered-X LMS algoritmussal [4].....	53
5.2.1. Analízis.....	55
5.3. On-line identifikáció rezonátoros rendszerekben.....	57
5.3.1 On-line identifikáció egysatornás rendszerre .....	58
6. A feladatmegoldás hardware és software eszközei .....	65
6.1. Az ADSP-21061 EZ-KIT Lite fejlesztőkártya.....	65
6.2. ADSP -21061 .....	65
6.2.1. Teljesítményadatok .....	66
6.2.2 Felépítés .....	66
6.2.3. Értékelés .....	68
6.3. A VisualDSP++ fejlesztői környezet .....	68
7. Megvalósított zajcsökkentő eljárások .....	71
7.1. Átvitelifüggvény mérése LMS algoritmussal .....	71
7.2. Átvitelifüggvény mérése rezonátoros struktúrával .....	72
7.3. Filtered-X LMS algoritmus .....	73
7.4. Filtered-EX LMS algoritmus .....	74
7.5. Unified-error Filtered-X LMS algoritmus.....	76
7.6. Rezonátoros zajcsökkentő struktúra.....	78
7.7. Rezonátoros zajcsökkentő struktúra on-line identifikációval .....	80
8. Teszteredmények.....	81
8.1. Mérési összeállítás.....	82
8.2. Filtered-X LMS algoritmus .....	83
8.3. Filtered-EX LMS algoritmus .....	88
8.4. Unified-error Filtered-X LMS algoritmus.....	93

8.5. Rezonátoros zajcsökkentő struktúra.....	98
8.6. Rezonátoros zajcsökkentő struktúra on-line identifikációval .....	101
9. Összefoglalás, kitekintés .....	105
10. Felhasznált irodalom .....	107

# 1. Absztrakt

A dolgozat bemutatja a legelterjedtebben használt aktív zajcsökkentő algoritmusokat, és azok on-line identifikációval való kiegészítését vizsgálja.

Aktív zajcsökkentésnek nevezzük az olyan eljárásokat, amelyek az alacsony frekvenciás zavarhatásokat a destruktív interferencia jelenségét kihasználva ellentétes fázisú zaj hozzáadásával szüntetik meg. Ez az elv már régóta ismert volt, de csak a digitális jelfeldolgozó processzorok elterjedésével vált megvalósíthatóvá. Az aktív zajcsökkentés a mérés technika egy tágabb szakterületének, az aktív inverse controlnak egy speciális és egyben legnehezebben megvalósítható területe.

Az aktív zajcsökkentés folyamata során szükségünk van az adott akusztikai rendszer bizonyos szintű identifikációjára. Az így alkotott modell beépül a zajcsökkentő eljárásba. Az akusztikus rendszerek azonban idővariáns rendszerek, ami miatt az identifikált modellünk érvényességét veszítheti, ami a teljes rendszer instabilitását eredményezheti. Ezért szükség van az akusztikus rendszer folyamatos, on-line identifikációjára.

Feladatom az volt, hogy áttekintsem az aktív zajcsökkentés irodalmát, különös tekintettel az akusztikus átvitel on-line identifikációjára, majd az irodalom alapján implementáljak már kifejlesztett, hatékony zajcsökkentő eljárásokat, valamint javaslatot tegyek a tanszéken periodikus jelek elnyomására kifejlesztett jelmodell alapú zajcsökkentő struktúra on-line identifikációval való kiegészítésének lehetőségeire és eredményeimet elméleti, szimulációs és mérési eredményekkel támasszam alá.

A diplomaterv bemutatja az aktív zajcsökkentésben legelterjedtebben használt algoritmusokat. A bemutatott eljárások két nagyobb, logikailag összetartozó csoportot alkotnak; az elsőt az LMS algoritmus alapú, a másodikat a periodikus jelmodell alapú zajcsökkentő struktúrák. Az LMS algoritmus alapú eljárások közül a Filtered-X, a Filtered- $\epsilon$ , a Filtered-EX és a Unified-error Filtered-X LMS algoritmus kerül bemutatásra. A dolgozat a fenti két csoport bemutatása után ezen algoritmusok on-line identifikációval való kiegészítésével foglalkozik, bemutatva már ismert, az irodalomban fellelhető módszereket az LMS algoritmus alapú struktúrákhoz, valamint egy önálló fejlesztésű algoritmust a periodikus jelmodell alapú zajcsökkentő struktúrához, melynek alapját egy, a Mérés technika és Információs Rendszerek Tanszéken felvetődött megoldási lehetőség képezte.

A zajcsökkentő eljárások bemutatása nem csupán elméleti síkon, hanem szimulációs eredményekkel alátámasztva történik. Az elméleti rész után a megvalósításhoz felhasznált hardware és software eszközök bemutatása következik. Ezt követi a megvalósított algoritmusok bemutatása. A dolgozatot a megvalósított algoritmusokkal végzett mérések és az így elért eredmények bemutatása zárja. A méréseket egycsatornás zajcsökkentő rendszerekkel végeztem. Az eredmények: periodikus jelek alapharmonikusára nézve 45 dB csillapítást, míg zajra 15 dB csillapítást értem el.



## ***1.1. Abstract***

The technology, which uses the phenomenon of destructive interference to suppress low- frequency acoustic disturbances with opposite phase noise, is called active noise control. Although this idea has been known for years, it became possible to realise only by the spreading of digital signal processors. Active noise control is a very special and not easily realisable application of the active inverse control.

A kind of identification of the acoustic system is needed during the active noise control process. The model created by this process integrates itself to the noise control system and it is essential for the stability of the system. These acoustic systems are time-variants so our model can loose it's validation, which can cause instability during the control process. Therefore the on-line identification of the acoustic system is needed to ensure the stability of the system.

My tasks were to skim and scan the literature of active noise control particularly the on-line identification of the acoustic systems, then according to the literature implement already known and effective noise control algorithms. Further task was to offer a suggestion to complete the resonator based noise control structure with on-line identification of the secondary path. And last, but not least I had to argue for my results with theoretical, simulation and measurement results.

The thesis presents the most popular algorithms used in active noise control, which can be divided into two major, logically connected parts. The first part consists of LMS algorithm based structures, and the second one periodical signal model based active noise control methods. Among the LMS algorithm based structures the Filtered-X, the Filtered- $\epsilon$ , on the Filtered-EX and the Unified-error Filtered-X LMS were presented in detail. After presenting these major parts of active noise control systems, the on-line identification of these algorithms were discussed, which were developed for the well-known LMS algorithm based structures. Additionally an individually developed algorithm related to periodical signal model structures was presented, which is based on an idea developed at the Department of Measurement and Information Systems.

The presentations of the active noise control systems confines themselves not only to academically facts, but confirmed by simulation results. The theoretical parts were followed by submit of the used hardware and software elements. After this the accomplished algorithms were shown. The thesis ended by testing the accomplished algorithms and submits the test results.

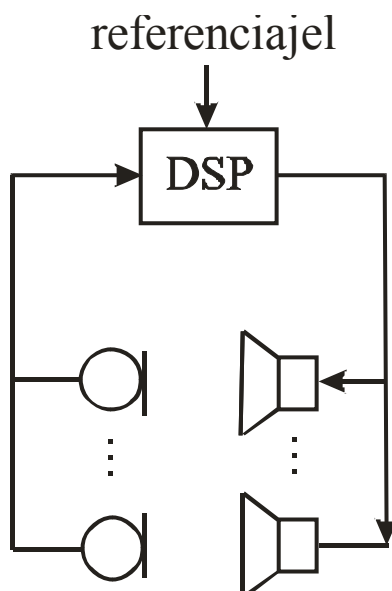


## 2. Bevezetés

A zajcsökkentés hagyományos módja, hogy a zajforrást és a kibocsátott zajtól megóvni kívánt területet fizikailag elválasztják egymástól különböző elnyelő anyagokkal. Ez történhet a zajforrás, vagy a megvédeni kívánt területet teljes vagy részleges elkerítésével. Ezeket passzív zajcsökkentő eljárásoknak nevezzük. Ezekkel az eljárásokkal nagy csillapítást érhetünk el, hátrányuk azonban, hogy meglehetősen költségigényesek, mivel használatukhoz speciális anyagokat kell alkalmazni, és az ilyen rendszerek telepítése (elrendezés megtervezése, az elnyelők megfelelő elhelyezése, rögzítése) igen nagy költségekkel jár. A passzív hangelnyelők másik hátránya, hogy sokszor elég nagy méretűek, amiből az következik, hogy szükség esetén elég nehezen mozgathatók, nem mobil berendezések. Ezen kívül ezek a rendszerek nem működnek jól alacsony frekvenciatartományban. Ennek az oka az, hogy például egy 100 Hz-es hang esetén a hullámhossz körülbelül 3,4 m (a levegőben, normál terjedési körülmények között). Ez a hullámhossz pedig igen nagy összemérve egy normál hangelnyelő felület vastagságával.

Ahol a passzív rendszerek már nem működnek olyan jól, ott lép színre az aktív zajcsökkentés (azaz az 1-2 kHz-ig terjedő tartományban) [9]. Az aktív zajcsökkentő rendszerek (ANC: Active Noise Control) tulajdonképpen elektromechanikus vagy elektroakusztikus rendszerek, amelyek a szuperpozíció elvén működnek. Ez azt jelenti, hogy a zavaró jelet (ami általában hallható zaj, de általánosabb értelemben lehet bármilyen rezgés) úgy szüntetik meg, hogy a zavarhatással egyező amplitúdójú, de ellentétes fázisú jelet juttatnak a fizikai rendszerbe (ahol a zavar fellép). Az általánosítást az indokolja, hogy például egy repülő szárnyának berezonálástól való mentesítése vagy egy szoba több pontjának zajmentesítése jelfeldolgozási szempontból nagyon hasonló feladat.

Egy aktív zajcsökkentő rendszer általános felépítését az 2.1. ábra mutatja.



2.1. ábra. Aktív zajcsökkentő rendszer általános felépítése

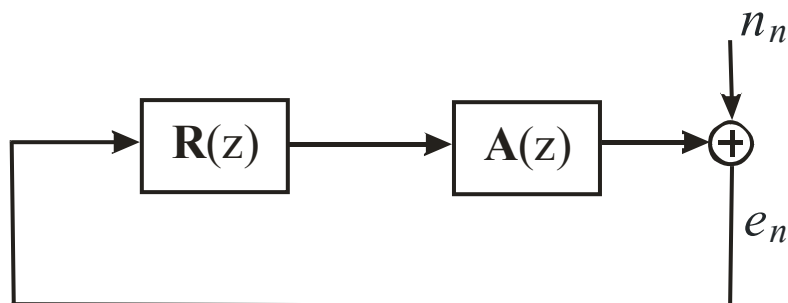
A zajforrásból érkező jelet az úgynevezett beavatkozó hangszórókból kiadott „inverz” zaj segítségével nyomjuk el, oly módon, hogy a térben a hanghullámok interferenciájának eredményeként az úgynevezett hibamikrofonok környezetében csöndes zónák alakuljanak ki.

Ezen csendes zónák a mikrofonok  $r$  sugarú környezetében alakulnak ki, ahol  $r$  körülbelül az elnyomandó jel hullámhosszának a negyede. Ez tulajdonképpen meg is szabja az ANC rendszerek korlátját, mivel nagyobb frekvenciákon a hullámhossz egyre kisebb, ami azt eredményezi, hogy a kialakítható csendes zónák mérete is jelentősen lecsökken.

A beavatkozó forrásokból kiadandó jelet a mikrofonok jeleinek és a zajforrásból például mikrofonnal vett úgynevezett referenciajel segítségével, digitális jelfeldolgozási algoritmussal határozhatjuk meg.

A feladat bonyolultsága miatt ANC rendszerek megvalósításához hagyományos mikrokontrollerek helyett jelfeldolgozó processzorokat alkalmaznak. Az ábrán ezért jelöltük úgy, hogy a mikrofonok és a beavatkozó hangszórók jelei között a digitális jelfeldolgozó processzor teremti meg a kapcsolatot. Ennek köszönhetően az aktív zajcsökkentő rendszerek diszkrét rendszerek, melyek az analóg fizikai világhoz AD- és DA-átalakítókön keresztül kapcsolódnak. Az egész rendszer vizsgálatához az analóg fizikai rendszert (átviteli közeg) is diszkrét modellel írjuk le. Ez a modellezés nem hibamentes, ezek a hibák azonban a gyakorlatban nem zavaróak. Feltételezhető továbbá, hogy a fizikai rendszer átviteli tulajdonságai időben változnak. A változás bekövetkezhet különböző okok miatt: emelkedik a hőmérséklet, változik a páratartalom, egy szoba esetében, ha kinyitják az ajtót. A fizikai rendszerek gyakran tartalmaznak nemlinearitásokat, melyeknek a kezelése nagyon nehéz. Szerencsére az akusztikus rendszerek igen széles tartományban lineárisnak tekinthetők. Az aktív zajcsökkentő rendszereket tehát lineáris időben variáns, diszkrét idejű rendszereknek tekinthetjük. Az akusztikus átvitel a mikrofonok és a hangszórók között a fent említett okok miatt változhat. Ez a változás pedig kedvezőtlen esetben a zajcsökkentő rendszer instabilitásához vezethet. Azért, hogy az akusztikus átvitel változásai miatt bekövetkező instabilitást elkerüljük, az akusztikus átvitelt folyamatosan ismernünk kell, azaz a rendszer on-line identifikációjára van szükség. Erre a feladatra még nincsenek általánosan használt, kiforrott megoldások. A fő problémát az jelenti, hogy az on-line identifikációt a zajcsökkentés megzavarása nélkül kellene végezni, ami azt jelenti, hogy identifikációs célra nagyon kicsi teljesítményű, vagy csak időnként hozzáadott gerjesztőjel alkalmazható, ami kedvezőtlenül befolyásolhatja az identifikáció sebességét, ami rossz esetben azt eredményezheti, hogy az identifikáció nem képes követni az akusztikus rendszer változásait. A legkedvezőbb megoldás zajcsökkentés szempontjából azonban az lenne, ha az identifikációt gerjesztés alkalmazása nélkül lehetne megoldani. A dolgozat ezekre a problémákra mutat be már meglévő, illetve új megoldásokat.

A fentieknek megfelelően a zajelnyomó rendszer modelljét mutatja az 2.2. ábra.



2.2 ábra. Aktív zajcsökkentő rendszer blokkvázlata

Az ábrán  $n_n$  az elnyomandó zaj,  $e_n$  a hibajel,  $\mathbf{R}(z)$  maga a zajcsökkentő struktúra,  $\mathbf{A}(z)$  pedig a fizikai rendszer átviteli függvénye abban az esetben, ha egy mikrofon és egy



hangszóró van. Más esetekben, azaz, ha több mikrofon és több hangszóró van, akkor  $A(z)$  egy átviteli függvény mátrix.

A zajcsökkentő eljárás célja tehát, hogy a hibajelet minimalizálja. A következő fejezetekben a tárgyalás egyszerűsítése érdekében egycsatornás rendszereket (egy mikrofon és egy hangszóró) vizsgálunk, hiszen ebben az esetben  $A(z)$  csak egy diszkrét idejű átviteli függvény, a zajcsökkentő struktúra kimenő jele skalár, ugyanígy a hibajel is. Természetesen az eredmények többcsatornás esetre is általánosíthatóak.

A következő két fejezetben a legfontosabb zajcsökkentő struktúrák kerülnek bemutatásra. Először az adaptív szűrővel felépített zajelnyomó rendszerek, valamint a nemrekurzív adaptív szűrők együtthatóinak gradiens alapú változtatását végző LMS (Least Mean Square) algoritmus, majd a jelmodell alapú zajelnyomó struktúrák ismertetése következik, amelyek periodikus zavaró jelek elnyomására nagyon hatékonyan alkalmazhatóak.

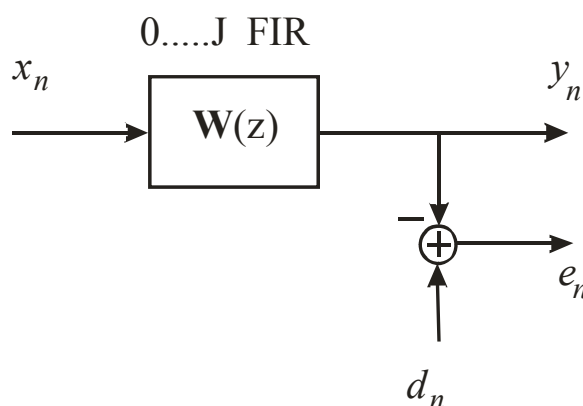


### 3. Zajcsökkentő eljárások LMS algoritmussal

Az aktív zajcsökkentő rendszerek működésének megismeréséhez nagyon fontos az adaptív szűrők elméletének áttekintése, hiszen ezek a rendszerek az adaptív szűrőket mint alapvető építőelemeket használják. Ezért ebben a fejezetben rövid áttekintést nyújtok a Wiener-szűrőkről, valamint az aktív zajcsökkentő rendszerekben használt LMS algoritmusról, majd bemutatom és elemzem a legfontosabb, és általam leghatékonyabbnak ítélt LMS algoritmussal működő zajcsökkentő struktúrákat.

#### 3.1. A Wiener-szűrők és az LMS algoritmus

A Wiener-szűrők speciális szűrők, amelyeket igen sok jelfeldolgozási alkalmazásban használnak (predikció, becslés stb.). Ezen szűrők esetén a szűrési feladat a következőképpen fogalmazható meg: a szűrőnek olyan együtthatókészlettel kell rendelkeznie, hogy a szűrő kimenete és a megkívánt jel különbsége négyzetének a várható értéke minimális legyen. A fentiek értelmezését a 3.1. ábra mutatja [7].



3.1. ábra. A Wiener-szűrő

Ahol:

$X_n$ : bemeneti sztochasztikus folyamat, melynek egy realizációja  $x_n$

$D_n$ : megkívánt sztochasztikus folyamat, melynek egy realizációja  $d_n$

$Y_n$ : kimeneti stochasztikus folyamat, melynek egy realizációja  $y_n$

Feltételek:

$X_n$  és  $D_n$  legalább gyengén stacionárius, és autokorrelációs függvényeik:

$$R_{XX}(m) = E\{X_m X_{n-m}\} \quad (3.1)$$

$$R_{DD}(m) = E\{D_m D_{n-m}\} \quad (3.2)$$

A keresztkorrelációs függvény:

$$R_{DX}(m) = E\{D_m X_{n-m}\} \quad (3.3)$$

A Wiener-feladat:

$$\min E\{(D_n - Y_n)^2\} = \min E\left\{(D_n - \sum_{j=0}^J w_j X_{n-j})^2\right\} \quad (3.4)$$

A megoldáshoz új jelöléseket érdemes bevezetni:

A szűrőegyütthatók vektora

$$\mathbf{w}^T = [w_0, w_1, \dots, w_J] \quad (3.5)$$

$$a_0 = R_{DD}(0) = E\{D_n D_n\} = E\{D_n^2\} \quad (3.6)$$

A keresztkorrelációs vektor:

$$\mathbf{b}^T = [b_0, b_1, \dots, b_J] \quad (3.7)$$

$$b_i = R_{DX}(i) = E\{D_n X_{n-i}\} \quad (3.8)$$

Az autokorrelációs mátrix:

$$\mathbf{R} = \begin{bmatrix} R_{00} & \cdot & R_{0J} \\ \cdot & R_{ij} & \cdot \\ R_{J0} & \cdot & R_{JJ} \end{bmatrix} \quad (3.9)$$

$$R_{ij} = R_{XX}(i-j) = R_{ji} \quad (3.10)$$

A fenti jelölésekkel a megoldás:

$$\min E\{(D_n - Y_n)^2\} = a_0 - 2\mathbf{w}^T \mathbf{b} + \mathbf{w}^T \mathbf{R} \mathbf{w} \quad (3.11)$$

A (2.11) egyenlet egy J+1 dimenziós térben elhelyezkedő paraboloid egyenlete. Ennek a minimuma ott van, ahol gradiense nullává válik, azaz:

$$2(\mathbf{R} \mathbf{w} - \mathbf{b}) = 0 \quad (3.12)$$

Így a Wiener-szűrő együtthatóinak optimális beállítására a következő képlet adódik:

$$\mathbf{w}_{opt} = \mathbf{R}^{-1} \mathbf{b} \quad (3.13)$$

A maradó hiba pedig ennek segítségével kifejezhető:

$$E\{(D_n - Y_n)^2\}_{opt} = a_0 - \mathbf{w}_{opt}^T \mathbf{b} \quad (3.14)$$

Az eddigiek alapján láthatjuk, hogy a Wiener-megoldás megkeresése tulajdonképpen szélsőérték-számítási feladat. A végeredménnyel azonban az a probléma, hogy az optimális szűrőegyütthatók kiszámításához a priori ismeretekkel kell rendelkezünk a bemeneti és a megkívánt jel statisztikai tulajdonságairól. Ezek a gyakorlatban nem állnak rendelkezésre, számításuk pedig nehézkes. Ezt a problémát hidalhatjuk át az LMS algoritmus

alkalmazásával, ami a nehézséget okozó statisztikai paraméterek helyett a pillanatnyi be- és kimenetből számított mennyiségeket felhasználva iteratív módon közelíti meg az optimális Wiener-megoldást. Így elkerülhetjük a bonyolult számítási eljárásokat, illetve megoldhatjuk az információhiányt. Ez azt jelenti, hogy az algoritmus az átlagos négyzetes hiba helyett a pillanatnyi hiba ( $e_n$ ) négyzetének várható értékét a minimalizálja.

Az LMS algoritmus tehát a negatív gradiens irányába lép oly módon, hogy a  $\mathbf{w}$  együtthatókkal rendelkező FIR szűrő együtthatóit módosítja, mégpedig a következőképpen:

$$e_n^2 = (d_n - y_n)^2 \quad (3.15)$$

$$(d_n - y_n)^2 = (d_n - \mathbf{w}_n^T \mathbf{x}_n)^2 \quad (3.16)$$

A pillanatnyi derivált ebben az esetben:

$$\frac{\partial e_n}{\partial \mathbf{x}_n} = -2(d_n - \mathbf{w}_n^T \mathbf{x}_n) \mathbf{x}_n \quad (3.17)$$

Azaz, a pillanatnyi deriváltban felismerve magát a hibát, ami rendelkezésre áll:

$$\frac{\partial e_n}{\partial \mathbf{x}_n} = -2e_n \mathbf{x}_n \quad (3.18)$$

Így az LMS módszer képlete:

$$\mathbf{w}_{n+1} = \mathbf{w}_n - \mu(-2e_n \mathbf{x}_n) \quad (3.19)$$

$$\mathbf{w}_{n+1} = \mathbf{w}_n + \mu 2e_n \mathbf{x}_n \quad (3.20)$$

ahol  $\mu$  az úgynevezett bátorsági tényező, melynek nagysága befolyásolja a lépésközt, és ezzel együtt a konvergenciasebességet. Optimális értéke az autokorrelációs mátrix ismeretében meghatározható. A következőkben megmutatom, hogy a hiba négyzetének várható értékét minimalizálva melyek az optimális  $\mathbf{w}$  együtthatók, és azt, hogy az együtthatókat LMS algoritmussal változtatva végül megkapjuk-e az optimális  $\mathbf{w}$  készletet. A hiba négyzetének várható értékét minimalizálva:

$$E\{e_n^2\} = E\left\{\left(d_n - \sum_{i=0}^J w_{n_i} x_{n-n-i}\right)^2\right\} \quad (3.21)$$

$$E\{e_n^2\} = E\left\{d_n^2 - 2 \sum_{i=0}^J w_{n_i} x_{n-n-i} + \sum_{i=0}^J w_{n_i} \sum_{j=0}^J w_{n_j} x_{n-n-i} x_{n-n-j}\right\} \quad (3.22)$$

A várhatóérték-képzést az összeg tagjaira külön-külön elvégezve, és bevezetve az autokorreláció ( $\mathbf{R}$ ) és a keresztkorreláció ( $\mathbf{b}$ ) fogalmát, az egyenlet a következőképpen alakul:

$$E\{e_n^2\} = E\{d_n^2\} - 2 \sum_{i=0}^J w_{n_i} b_i + \sum_{i=0}^J w_{n_i} \sum_{j=0}^J w_{n_j} R_{i-j} \quad (3.23)$$

A fenti egyenletet mátrixos formába átírva látható, hogy az egyenlet tulajdonképpen egy kvadratikus alak, azaz a hibafelület egy paraboloid a paraméterek síkja fölött. Az egyenlet mátrixos alakban ( $E\{d(n)^2\} = a$  helyettesítéssel):

$$E\{e_n^2\} = a_0 - 2\mathbf{w}^T \mathbf{b} + \mathbf{w}^T \mathbf{R} \mathbf{w} \quad (3.24)$$

Az optimális  $\mathbf{w}$  együtthatókat a fenti kifejezés minimuma adja, ehhez deriválni kell a kifejezést  $\mathbf{w}$  szerint, majd az eredményt egyenlővé téve nullával, a végeredmény, azaz az optimális  $\mathbf{w}$  együtthatók:

$$\mathbf{w}_{opt} = \mathbf{R}^{-1} \mathbf{b} \quad (3.25)$$

Most vizsgáljuk meg, hogy az LMS algoritmus a pillanatnyi hiba minimalizálásával végül az optimális  $w$  készletet határozza-e meg. Ha az LMS algoritmus beért az optimumba, akkor:

$$E\{\mathbf{w}_{n+1}\} = E\{\mathbf{w}_n\} \quad (3.26)$$

Ekkor az LMS algoritmus képlete alapján:

$$E\left\{\left(d_n - \sum_{i=0}^J w_{n_i} x_{n-n-i}\right) \mathbf{x}_n\right\} = 0 \quad (3.27)$$

$$E\{d_n \mathbf{x}_n\} = \sum_{i=0}^J w_i E\{x_{n-k} x_{n-i}\} \quad (3.28)$$

$$\mathbf{b} = \mathbf{R} \mathbf{w} \quad (3.29)$$

Ahogy a fenti egyenlet mutatja, az LMS algoritmus az optimális állapot felé konvergál. Tehát az LMS algoritmus az autokorrelációs mátrix és a keresztkorrelációs vektor ismerete nélkül képes megtalálni azokat a  $w$  együtthatókat, melyekkel az adaptív szűrő kimenete a lehető legjobban „hasonlít” a megkívánt jelre. Ez azért nagyon fontos, mivel általában a gyakorlati alkalmazások során sem az autokorrelációs mátrix, sem a keresztkorreláció nem ismert. Az LMS algoritmus viszont nagyon jól alkalmazható abban az esetben, ha valós időben ismert az adaptív szűrő kimenete és a megkívánt jel közti különbség. Az aktív zajcsökkentő eljárások ilyenek, hiszen az a cél, hogy az elnyomandó jellel azonos jelet adjunk ki (csak ellentétes fázisban), a hibamikrofonnal pedig azt a jelet vesszük, ami az LMS algoritmus számára szükséges. Ezért alkalmazzák előszeretettel az LMS algoritmust és különböző változatait az aktív zajcsökkentő eljárásokban.

A következőkben bemutatásra kerül, hogy mi a hatása annak, hogy az LMS algoritmus nem a pontos gradiens vektor, hanem tulajdonképpen annak becslését használja az adaptációhoz. Ez azt jelenti, hogy az igazi gradiens vektorhoz zaj adódik:

$$\hat{\mathbf{V}} = -2e_n \mathbf{x}_n = \nabla_n - \mathbf{n}_n \quad (3.30)$$

Ha feltesszük, hogy az adaptív szűrő együtthatói már közel járnak a Wiener-megoldáshoz, akkor kijelenthetjük, hogy mivel az igazi gradiens nulla, és mi mégis becsülni próbáljuk az LMS algoritmus képlete alapján, a gradiens vektor zaja egyenlő lesz a becsült értékkel, ahogy az alábbi képlet mutatja:

$$\mathbf{n}_n = 2e_n \mathbf{x}_n \quad (3.31)$$

A fenti képlet alapján belátható, hogy a gradiens zaj kovarianciája a következő:

$$\text{cov}\{n_n\} = 4E\{e_n^2\} \mathbf{R} \quad (3.32)$$

Ha feltesszük, hogy az adaptív szűrő együtthatói közel járnak a Wiener-megoldáshoz, akkor a pillanatnyi hiba négyzetének várható értéke megközelítőleg megegyezik a Wiener-megoldás minimális hibájával, így a képlet a következőképpen alakul:

$$\text{cov}\{n_n\} = 4e_{\min}\mathbf{R} \quad (3.33)$$

A gradiens zaj azért fontos, mivel tovaterjedve zajt okoz a szűrőegyütthatókban, ami azt eredményezi, hogy a szűrő kimenetén a hibajel négyzetének várható értéke nagyobb lehet, mint ami optimális együttható-beállítással elérhető. Az adaptáció konvergenciája után megmaradó átlagos négyzetes hiba és a minimális hiba különbségének és a minimális hibának a hányadosát nevezik angolul „misadjustment”-nek, és  $m$ -el jelölik:

$$m \triangleq \frac{e^+}{e_{\min}} \quad (3.34)$$

Ahol  $e^+$  a minimális hibától való átlagos eltérést jelöli. Ha  $m$  értéke például 10%, akkor az azt jelenti, hogy az adaptív rendszer átlagos középhibája mindössze 10%-kal nagyobb az optimális beállítással elérhetőnél (Wiener-megoldás). Az  $m$  értékét kis eltérésekre a következőképpen becsülhetjük:

$$m = \mu \text{tr}(\mathbf{R}) \quad (3.35)$$

Az előbbi képlet levezetése során tett feltevések miatt a képlet akkor ad jó becslést, ha a szűrőegyütthatók értéke „jól közelíti” a Wiener-megoldást, azaz körülbelül 25% alatti  $m$  értékekre. A képlet egy másik érdekes alakja származtatható, ha feltesszük, hogy az autokorrelációs mátrix sajátértékei egyformák. Ekkor  $m$  kifejezhető az adaptív szűrő együtthatószámával ( $n$ ) és az LMS algoritmus adaptációs idejével ( $\tau$ ):

$$m = \frac{n}{4\tau} \quad (3.36)$$

A fentiek alapján belátható, hogy az LMS algoritmussal nem érhetjük el az optimális beállítást, de tetszőlegesen közel kerülhetünk hozzá. Ez a hiba a gyakorlati alkalmazásokban nem okoz problémát, sokszor elhanyagolható az egyéb más eszközök által okozott hiba mellett.

### 3.1.1. Az akusztikus átvitel mérése

Ahogy a 2.2. ábrán láttuk, az aktív zajcsökkentő rendszerekben jelen van egy akusztikus átvitel a beavatkozó hangszórók és a mikrofonok között. Ez egyszerű, egy csatornás, azaz egy hibamikrofont és egy beavatkozó hangszórót alkalmazó rendszerek esetén egy átviteli függvény. Az ANC rendszerek legtöbbjének működéséhez ezt az átvitelt ismernünk kell bizonyos pontossággal, tehát valamilyen módon identifikálnunk kell azt. Az átviteli függvény mérése általános feladat, így erre több megoldás is kínálkozik. Az alábbiakban ezek közül sorolok fel néhányat:

- multiszínuszos gerjesztés, kiértékelés FFT-vel
- fehér zaj gerjesztés, kiértékelés FFT-vel
- adaptív szűrő (általában FIR típusú) alkalmazása az átviteli út modellezésére

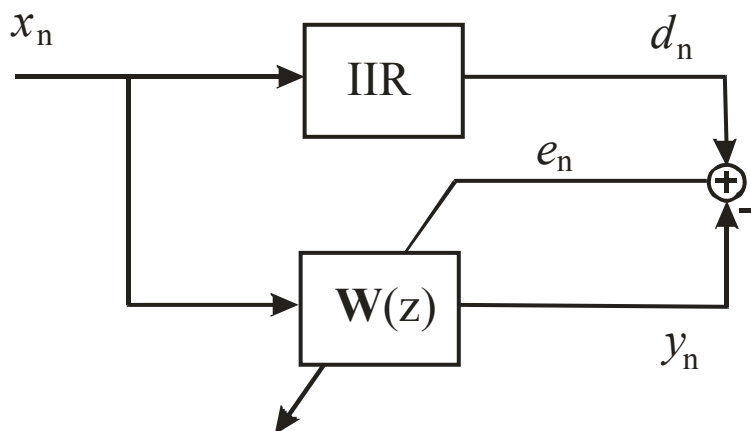
Az első esetben a mérés varianciája kicsi, az eljárás gyors, de az optimális gerjesztőjel megkonstuálása nehézkes, iteratív úton történhet. Ezen kívül multiszínusz gerjesztés esetén egy szinuszjelre nagyon kis teljesítmény jut. További hátrány, hogy  $A(z)$ -nek, amely az akusztikus rendszeren keresztül analóg elektronikát is tartalmaz, intermodulációs torzítása is lehet, emiatt egy adott frekvencián a kis amplitúdójú hasznos jelhez nagyobb amplitúdójú lekeverődő zavaró jel adódik, ezzel elrontva a mérést.

A második módszer jól alkalmazható, hátránya a mérés nagy varianciája, valamint az is elképzelhető, hogy azokon a frekvenciákon, ahol  $A(z)$  átvitele kicsi, esetleg nem gerjeszti elég jól a rendszert.

Ha azonban maga a zajcsökkentő rendszer adaptív szűrő alapú, akkor legkézenfekvőbb módszerként az LMS algoritmussal történő identifikáció kínálkozik. Ez tulajdonképpen a második módszer egy megvalósítása, nagyon pontos eredményt ad, de természetesen az így kapott modell különbözni fog a valódi átviteltől. Ennek a különbözőségnek a forrásai a következők lehetnek [1]:

- a különbség első forrása, hogy egy végtelen impulzusválaszú rendszert modellezünk egy FIR, azaz véges impulzusválaszú rendszerrel
- a második lehet magának a gerjesztőjelnek a tökéletlensége, azaz ha nem alkalmazunk megfelelő gerjesztést, hibás eredményt kaphatunk
- a hiba harmadik forrása az lehet, hogy mivel az adaptáció LMS algoritmussal történik, a Wiener-megoldással elérhető minimális hibához a korábbiakban leírtak szerint hozzáadódik egy többlet hiba ( $m$ )

A fentiek alapján látható, hogy lesz bizonyos modellezési hiba, aktív zajcsökkentő alkalmazások esetén azonban a másodlagos átviteli utat nem szükséges nagy pontossággal ismernünk, a másodlagos út modelljének fázishibája nem haladhatja meg a  $\pi/2$  értéket, és a fenti forrásokból eredő hiba olyan kis mértékű is lehet, hogy azzal ANC alkalmazásokban nem kell számolnunk. A harmadik módszert választottam tehát, mivel pontossága kielégítő jelen alkalmazásban, valamint az adaptáció után kapott szűrőegyütthatók közvetlenül felhasználhatóak LMS algoritmussal működő aktív zajcsökkentő alkalmazásokban. Ennek a mérési, identifikációs módszernek a blokkvázlata látható a 3.2. ábrán:



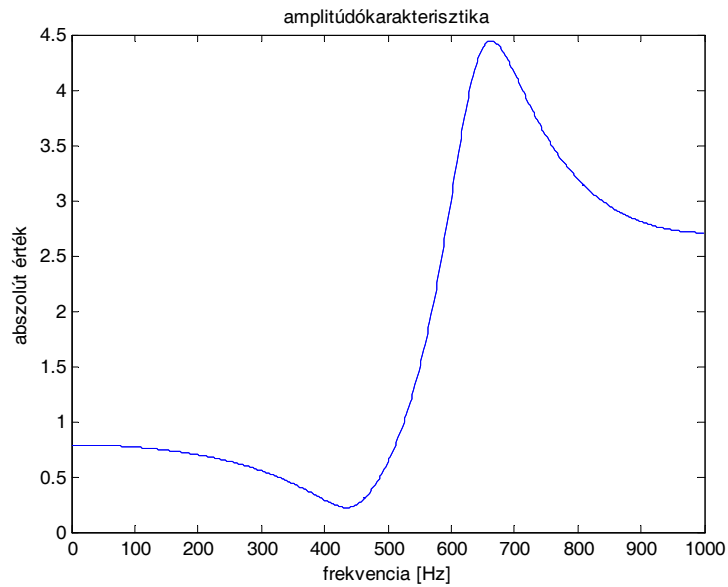
3.2. ábra. Átvitel mérése LMS algoritmussal



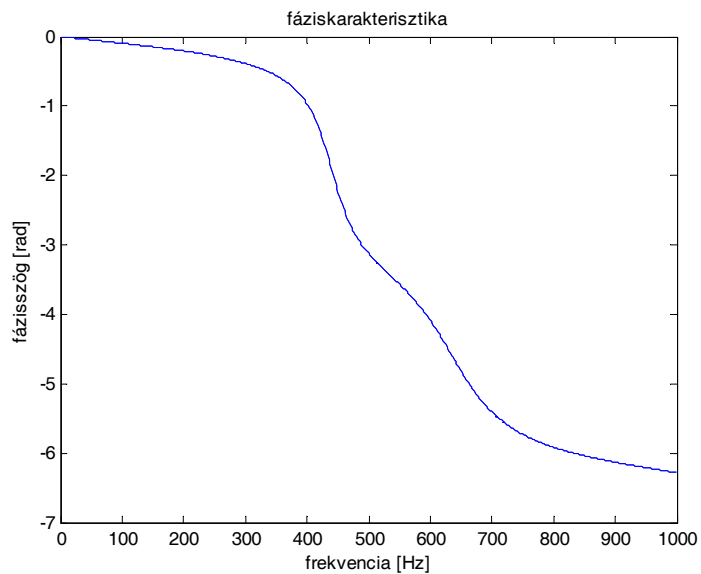
MATLAB-ban megvalósítottam a 3.2. ábrán látható mérési elrendezés szimulációját, a következőkben pedig bemutatom a szimulációs eredményeket. A szimuláció során az identifikálandó átviteli függvény a következő képlettel volt adott:

$$W(z) = \frac{z^2 - 0.4164z + 1.2346}{z^2 + 0.6627z + 0.6414} \quad (3.37)$$

A fenti képlettel adott IIR rendszer amplitúdó- és fáziskarakterisztikája látható a 3.3. és a 3.4. ábrán:



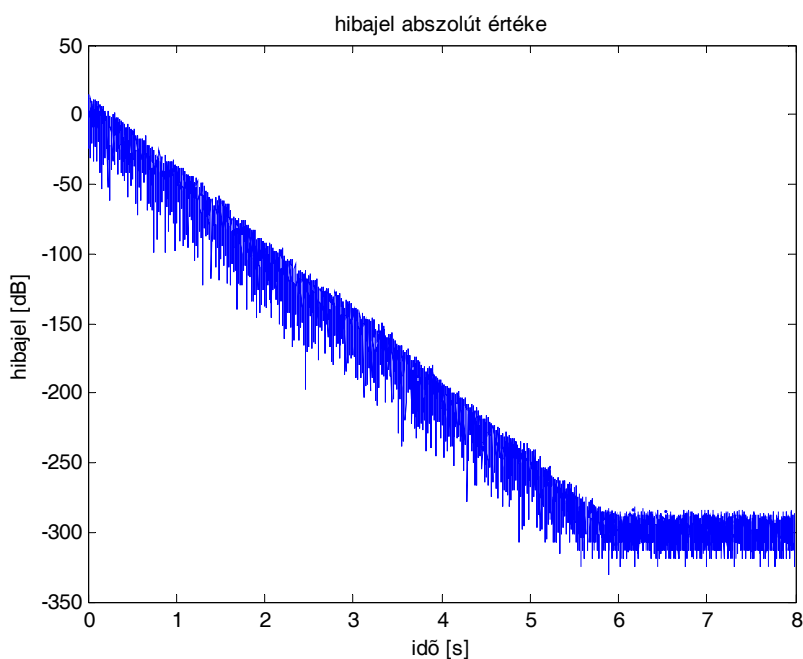
3.3. ábra. Identifikálandó rendszer amplitúdókaraktisztikája; mintavételi frekvencia 2 kHz



3.4. ábra. Identifikálandó rendszer fáziskarakterisztikája; mintavételi frekvencia 2 kHz

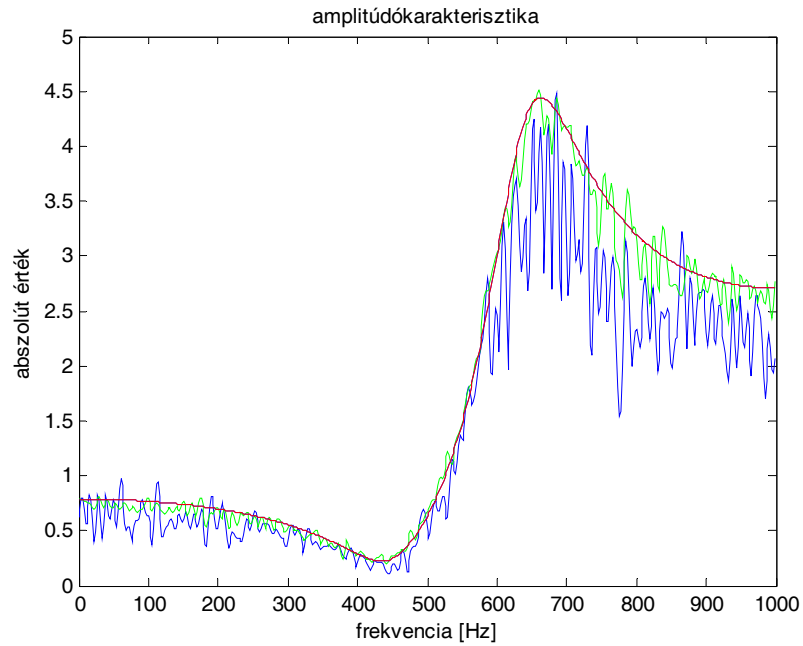
A fenti átviteli függvény nem hasonlít az aktív zajcsökkentési gyakorlatban mérhető átviteli karakterisztikákhoz, szimulációban való használatát ennek ellenére az indokolja, hogy viszonylag nagy dinamikával rendelkező, nem csupán néhány egységnyi késleltetést biztosító, szimulációs környezetben mégis egyszerűen megvalósítható szűrőről van szó, ami az akusztikus átvitelhez hasonlóan jól szelektál az algoritmusok között.

A szimuláció során 2 kHz-es mintavételi frekvenciát feltételeztem, a mérési idő 8 s, azaz 16000 minta volt. Az adaptív szűrő együtthatószáma 200, a bátorsági tényező  $\mu = 0.002$ . A fenti beállítással az identifikáció során a hibajel alakulása látható a 3.5. ábrán. Az ábrán a hibajel logaritmikusan ábrázoltam, ezért jól látható, hogy körülbelül 6 s után a hibajel már nem csökken tovább, megkapjuk a maradó hibát.



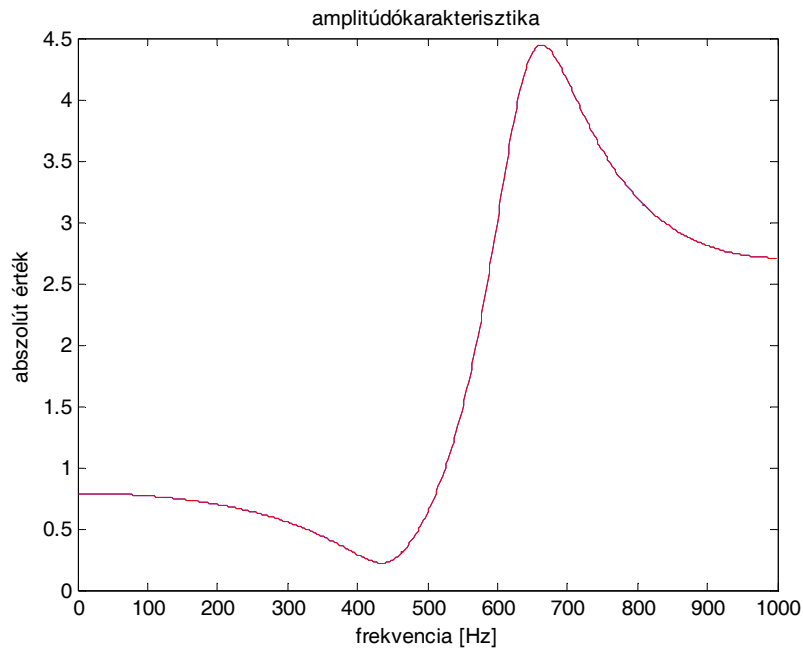
3.5. ábra. A modellezési hiba alakulása az idő függvényében; mintavételi frekvencia 2 kHz

A következő ábrán bemutatom, hogyan változik az adaptív szűrő amplitúdókarakterisztikája az idővel. A 3.6. ábrán látható az adaptív szűrő amplitúdókarakterisztikájának változása. Kék színnel jelöltem a 400 minta, azaz 0.2 s utáni, zölddel a 800 minta, azaz 0.4 s utáni, és pirossal a 16000 minta, azaz 8 s utáni állapotot, és végül feketével az eredeti átviteli függvényt. Az ábrán jól látható, hogy az identifikáció utáni átviteli függvény már teljesen rásimul az eredetire, eltakarva azt.

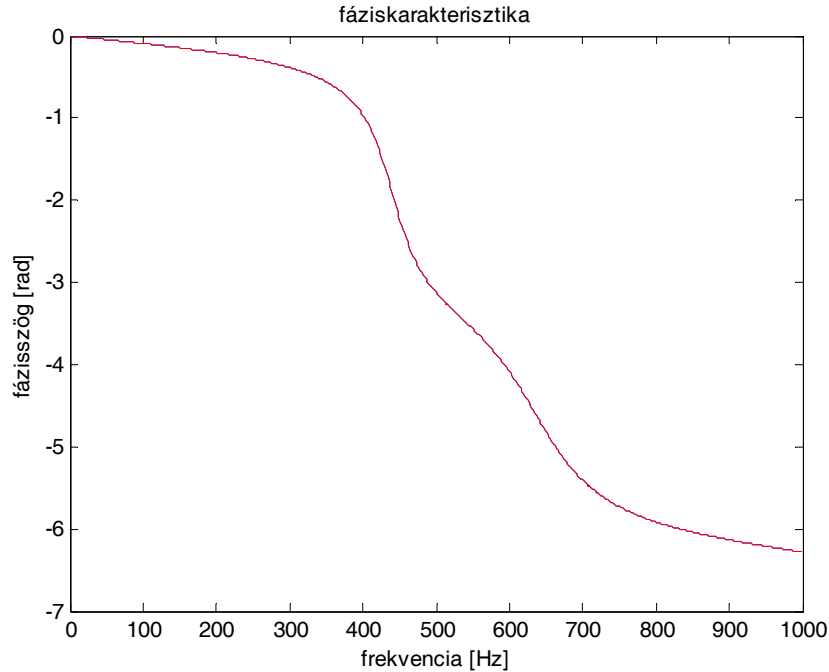


3.6. ábra. Az adaptív szűrő amplitúdókarakterisztikájának alakulása; mintavételi frekvencia 2 kHz

A 3.7. és 3.8. ábrán látható az identifikáció után kapott amplitúdó és fáziskarakterisztika, ezt pirossal jelöltem az ábrákon, míg az eredeti karakterisztikát késsel. Az ábrákról jól látszik, hogy az identifikált átviteli függvény teljesen rásimul az eredetire, a mérés tehát sikeres volt.



3.7. ábra. Az adaptív szűrő amplitúdókarakterisztikája identifikáció után; mintavételi frekvencia 2 kHz

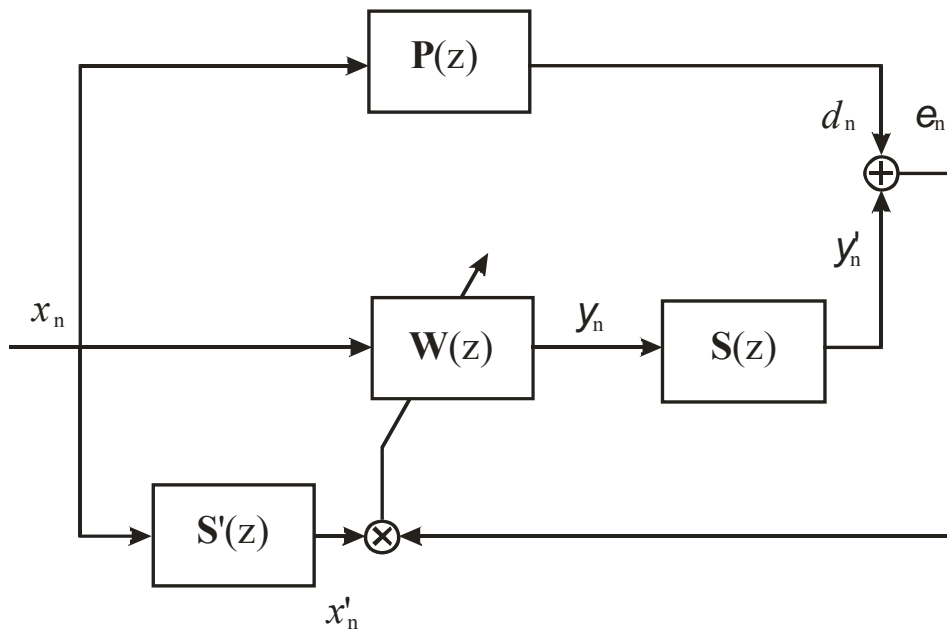


3.8. ábra. Az adaptív szűrő fáziskarakterisztikája identifikáció után; mintavételi frekvencia 2 kHz

A szimulációk során meggyőződtem arról, hogy a választott mérési eljárás a vártnak megfelelően működik, és gyakorlatot szereztem a paraméterek megfelelő beállításában.

### 3.2. *A Filtered-X LMS algoritmus [1]*

Aktív zajcsökkentő alkalmazásokban számolnunk kell a beavatkozó hangszórók és a hibamikrofonok közötti másodlagos út átvitelével. Ezért ilyen alkalmazásokban az egyszerű LMS algoritmus nem használható, mivel a másodlagos út okozta fázistolás miatt az algoritmus nem konvergál, instabil lesz. Ennek a problémának a megoldására született meg a Filtered-X LMS eljárás. Az algoritmushoz szükség van egy modellre a másodlagos út átviteléről. Ez a modellezés off-line elvégezhető az egyszerű LMS algoritmus használatával, ahogyan azt a 3.1.1-es fejezetben bemutattam. A Filtered-X LMS algoritmus blokkvázlata a 3.9. ábrán látható. Az ábrán  $P(z)$  jelöli az elsődleges utat (primary path),  $S(z)$  a másodlagos utat (secondary path),  $S'(z)$  pedig a másodlagos átviteli út off-line meghatározott modelljét.

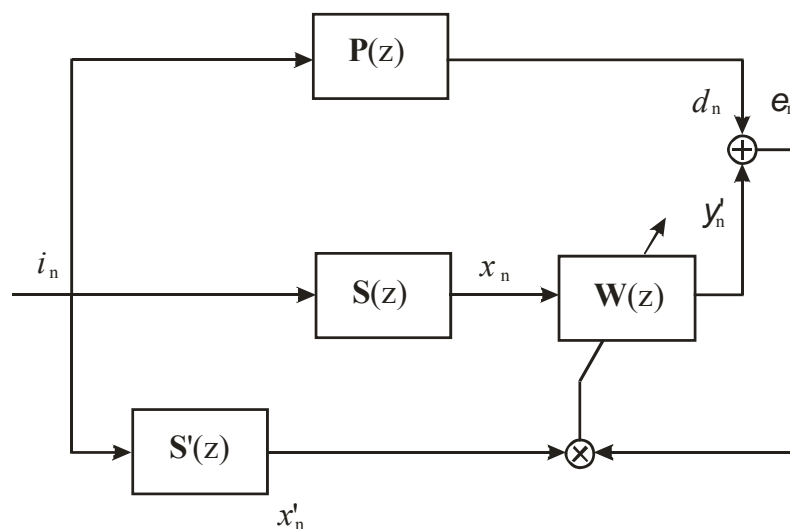


3.9. ábra. A Filtered-X LMS algoritmus blokkvázlata

A következő fejezetben az imént bemutatott algoritmus analízisét végzem el, megmutatom, hogy  $W(z)$  adaptálásával az együtthatók a fenti problémára vonatkozó Wiener-megoldáshoz konvergálnak, majd a módszer működésének igazolására szimulációs eredményeket mutatok be.

### 3.2.1. A Filtered-X LMS algoritmus analízise

Az analízis során egyszerűsítésképp felteszem, hogy  $W(z)$  impulzusválasza nem véges. Mivel a rendszer lineáris, ezért  $W(z)$  és  $S(z)$  felcserélhető, így a 3.9. ábra a következőképp rajzolható át [1]:



3.10. ábra. Filtered-X LMS algoritmus analízise

Ekkor felhasználva azt, hogy egy Wiener-szűrő bemenete, és a hibajel nem korrelált egymással, tehát:

$$E\{x'_k e_{k+m}\} = 0 \quad (3.38)$$

A fenti képletet tovább vezetve:

$$E\{x'_k (d_{k+m} - y'_{k+m})\} = 0 \quad (3.39)$$

$$E\left\{x'_k d_{k+m} - \sum_{l=-\infty}^{\infty} w_l x_{k-l+m} x'_k\right\} = 0 \quad (3.40)$$

Amiből Z-transzformáció után:

$$W(z) = \frac{R_{x'd}(z)}{R_{x'x}(z)} \quad (3.41)$$

A 3.10. ábra alapján felírható, hogy:

$$R_{x'd}(z) = R_{ii}(z)S'(z^{-1})P(z) \quad (3.42)$$

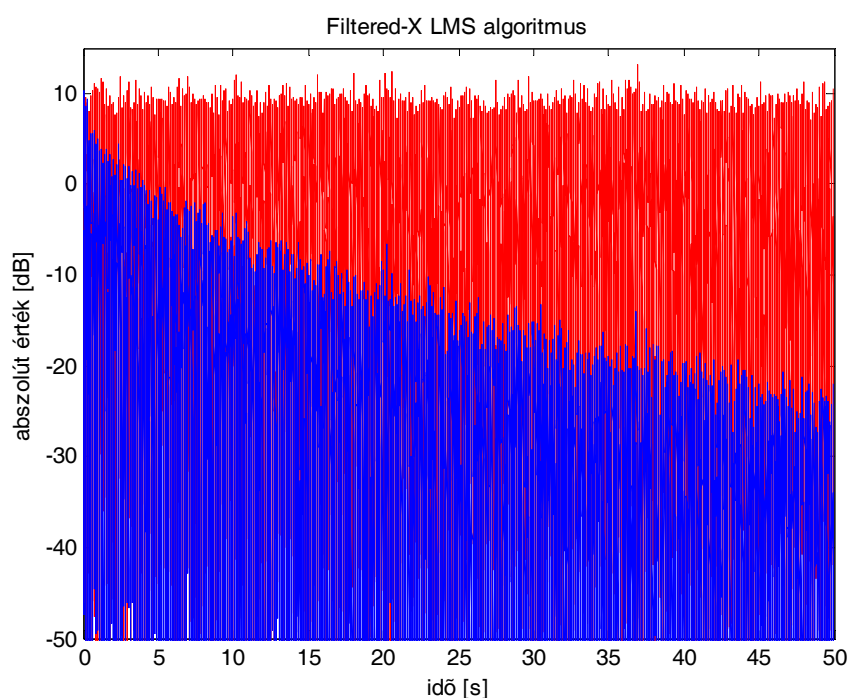
$$R_{x'x}(z) = R_{ii}(z)S'(z^{-1})S(z) \quad (3.43)$$

Amiből a végeredmény:

$$W(z) = \frac{P(z)}{S(z)} \quad (3.44)$$

Ez pedig a Wiener-megoldás! Tehát  $W(z)$  adaptációja a Wiener-megoldásra vezet, és  $S'(z)$ -nek nincs hatása a megoldásra, tehát az csak a rendszer stabilitásában játszik szerepet. A gyakorlatban természetesen nem érhető el a tökéletes Wiener-megoldás, hiszen  $W(z)$  impulzusválasza véges.

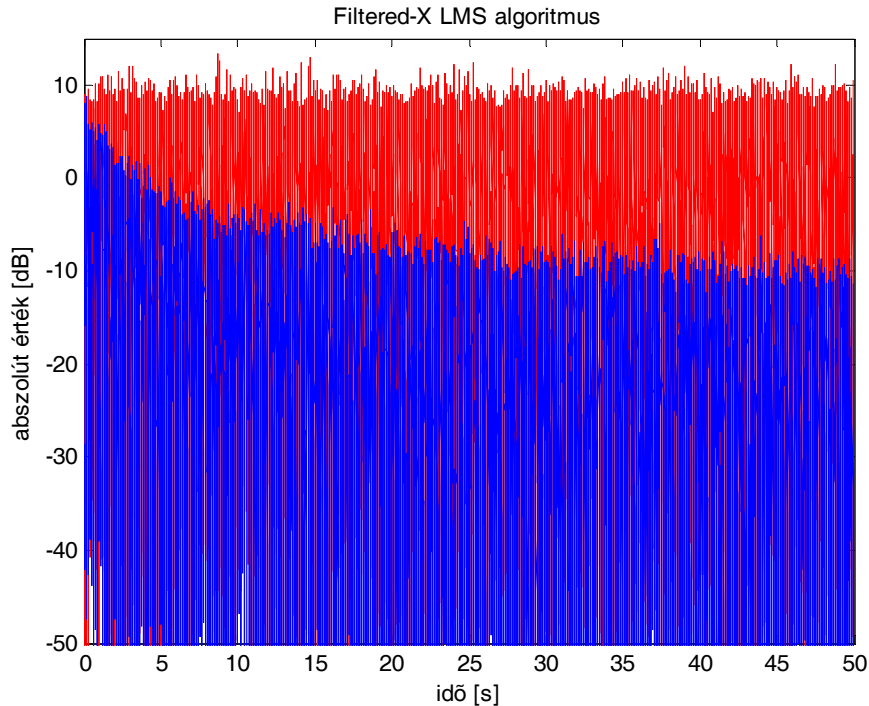
Az elméleti eredményeket szimulációval támasztottam alá. A szimulációban az elnyomandó jel normális eloszlású, zérus várható értékű, egységnyi varianciájú zaj. Az alkalmazott mintavételi frekvencia 2 kHz. Az elsődleges utat egy egyszerű egységnyi átvitelű, 50 késleltetésű FIR szűrővel modelleztem, míg a másodlagos út a (3.37) egyenlettel adott IIR szűrő volt. A másodlagos út modellezését off-line végeztem egy 200 együtthatós FIR szűrővel, az előző fejezetben megadott módon. Az alkalmazott  $W(z)$  szűrő szintén 200 együtthatós, a bátorsági tényező  $\mu=0.0002$  volt. Ennél nagyobb bátorsági tényezőt alkalmazva a rendszer nem bizonyult stabilnak ilyen beállítások mellett. Így az elnyomásra a 3.11. ábrát kaptam. Az ábrán pirossal jelöltem az elnyomás nélküli, kézzel az elnyomással kapott állapotot (maradó hibajel).



3.11. ábra. Filtered-X LMS algoritmus működése; mintavételi frekvencia 2 kHz

Az ábráról megállapítható a Filtered-X LMS algoritmus legfőbb hátránya, a lassú beállítás, azaz kicsi konvergenciasebesség. Ennek javítására természetesen sok megoldás született, a következőkben ezek közül be is mutatok néhányat.

A fenti szimulációs modell azonban nem veszi figyelembe azt, hogy a gyakorlatban a hibamikrofonnál az elnyomandó zajhoz a referencijellel nem korrelált zajok is adódnak. Ezek nem befolyásolják a rendszer konvergenciáját, mivel a (3.39) képlet szerint a referencijellel nem korrelált zajok a várhatóérték-képzés miatt kiesnek, tehát a  $W(z)$  szűrő konvergenciáját nem befolyásolják; mivel azonban a referencijellel nem korreláltak, nem kerülnek elnyomásra, értékük a maradó hiba értékét növeli. Erre láthatunk szimulációs példát a 3.12. ábrán, ahol a hibamikrofon jeléhez a referencijellel nem korrelált, nulla várható értékű, de a referencijelnél sokkal kisebb varianciájú zaj adódik. A szimuláció egyéb beállításai megegyeznek az előző esetben alkalmazottakkal.



3.12. ábra. Filtered-X LMS algoritmus működése nem korrelált zaj jelenlétében; mintavételi frekvencia 2 kHz

A fentiek alapján a Filtered-X LMS algoritmus jól használható, egyszerű, robusztus algoritmus, ami nagyon jól használható aktív zajcsökkentő alkalmazásokban. Hátránya, hogy konvergenciasebessége igen kicsi. A továbbiakban ezért olyan eljárásokat vizsgáltam, amelyek megtartják a Filtered-X LMS algoritmus jó tulajdonságait nagyobb konvergenciasebesség mellett. Ez természetesen azzal jár, hogy az eljárások bonyolultabbak lesznek, nagyobb számítási kapacitást igényelnek.

### 3.3. A Filtered- $\varepsilon$ LMS algoritmus [1]

A Filtered- $\varepsilon$  LMS algoritmus az előző fejezetben tárgyalt Filtered-X LMS algoritmus továbbfejlesztéseként született. A legfőbb különbség a kettő között az, hogy a Filtered- $\varepsilon$  LMS algoritmus nem csupán a referenciajelet, hanem a hibamikrofon jelét is megszüri, és ezzel a két jellel végzi a  $W(z)$  szűrő adaptációját. Természetesen ehhez az algoritmushoz is szükség van a másodlagos út modelljére, ami off-line meghatározható az egyszerű LMS algoritmussal. Ezt követően a másodlagos út modelljének segítségével meghatározzuk a másodlagos út inverzét szintén off-line.

Ha a másodlagos út minimálfázisú, akkor ez az inverz egyszerűen előállítható. Vegyük például a következő átviteli függvényt:



$$S(z) = \frac{1 + \frac{1}{2}z^{-1}}{1 - z^{-1} + \frac{3}{4}z^{-2}} \quad (3.45)$$

A fenti átviteli függvény inverze, mivel a függvény minimálfázisú, ezért stabil lesz, és a következőképp áll elő:

$$S^{-1}(z) = \frac{1 - z^{-1} + \frac{3}{4}z^{-2}}{1 + \frac{1}{2}z^{-1}} \quad (3.46)$$

A fenti függvény ki is fejthető polinom osztás segítségével, az eredmény pedig a következő[1]:

$$S^{-1}(z) = 1 - \frac{3}{2}z^{-1} + \frac{3}{2}z^{-2} - \frac{3}{4}z^{-3} + \frac{3}{8}z^{-4} - \dots \quad (3.47)$$

A képlet alapján jól látható, hogy az inverz stabil és kauzális lett. Az aktív zajcsökkentési gyakorlatban azonban a másodlagos út átvitele tipikusan nem minimálfázisú. A feladat enne ellenére az, hogy előállítsuk a megfelelő inverz átvitelt. Ezt a következőképpen lehet megkonstruálni. Vegyünk egy nem minimálfázisú átviteli függvényt:

$$S(z) = \frac{1 + 2z^{-1}}{1 - z^{-1} + \frac{3}{4}z^{-2}} \quad (3.48)$$

Ekkor az inverz:

$$S^{-1}(z) = \frac{1 - z^{-1} + \frac{3}{4}z^{-2}}{1 + 2z^{-1}} \quad (3.49)$$

A (3.49) egyenlettel adott átviteli függvény nem stabil, mert pólusai kívül esnek az egységkörön.  $S^{-1}(z)$  azonban kifejthető az előbbieken említett polinomosztással kétféleképpen [1]:

$$S^{-1}(z) = 1 - 3z^{-1} + \frac{27}{4}z^{-2} - \frac{27}{2}z^{-3} + 27z^{-4} + \dots \quad (3.50)$$

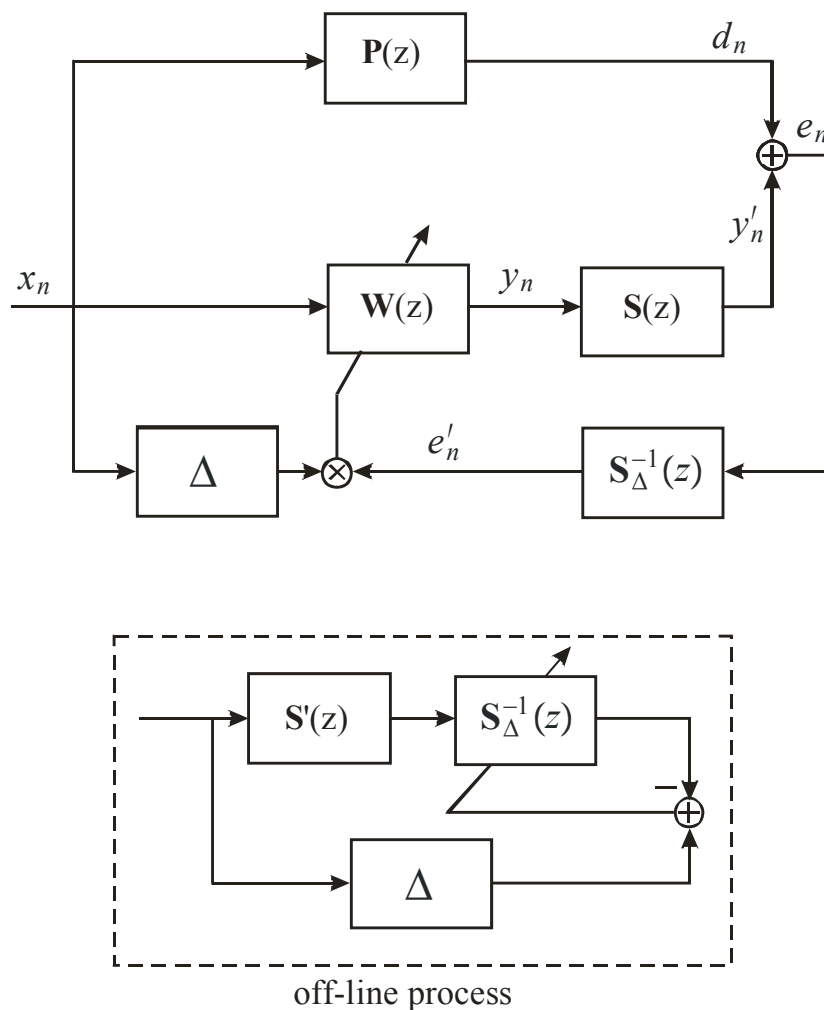
$$S^{-1}(z) = \frac{3}{8}z^{-1} - \frac{11}{16} + \frac{27}{32}z + \frac{27}{64}z^2 + \frac{27}{128}z^3 + \dots \quad (3.51)$$

A (3.50) képlettel adott kifejtés kauzális, de nem stabil, míg a (3.51) képlettel adott megoldás stabil, de nem kauzális. Ezen megoldások közül egyik sem használható. A (3.50) képlettel adott megoldás az MSE (Mean Square Error) értékét végtelenbe növelné. A (3.51) képlet szerinti megoldás viszont nem kauzális, tehát nem realizálható, kivéve az összeg első két tagját, mivel azok megvalósíthatóak. Ha a maradék tagok az összegben elhanyagolhatóan kicsik lennének, akkor az ily módon megvalósított szűrő a nem minimálfázisú rendszer

inverzének egy jó közelítését adná. Ezt az ötletet tovább vezetve, ha  $S(z)$  és  $S^{-1}(z)$  szorzásának eredményeképpen késleltetést is megengedünk, akkor IIR szűrőt feltételezve az ily módon előállított inverz pontossága tetszőlegesen növelhető a késleltetés növelésével. FIR rendszerre ez már nem igaz, ott meg kell találni az optimális késleltetést, mivel annak növelése egy pont után már rontja a pontosságot. Ennek az ötletnek az alkalmazásával a 3.13. ábra alsó részén látható módon valósítható meg a késleltetett inverz előállítása adaptív modellezéssel.

A fent leírt eljárás nagyon hasznos, ugyanakkor a megvalósítása aktív zajcsökkentő eljárásokban körülményes, mivel ANC alkalmazásokban a másodlagos átviteli utat tipikusan FIR szűrővel modellezzük. Ez pedig a fentebb leírtak alapján azt jelenti, hogy minden egyes átviteli függvény-mérés után meg kell határozni egy optimális késleltetést (továbbiakban  $\Delta$ ), hogy a késleltetett inverz megfelelő pontosságú legyen, és  $\Delta$  értékét nem választhatjuk tetszőlegesen nagyra, ami azért lenne jó, mivel igen nagy  $\Delta$ -t választva elég lenne csak egyszer kiválasztani egy megfelelően nagy  $\Delta$ -t (mivel ez FIR rendszer esetén a fent leírtak miatt káros lehet).

A Filtered- $\epsilon$  LMS algoritmus, és az off-line inverz modellezés blokkvázlata látható a 3.13. ábrán.

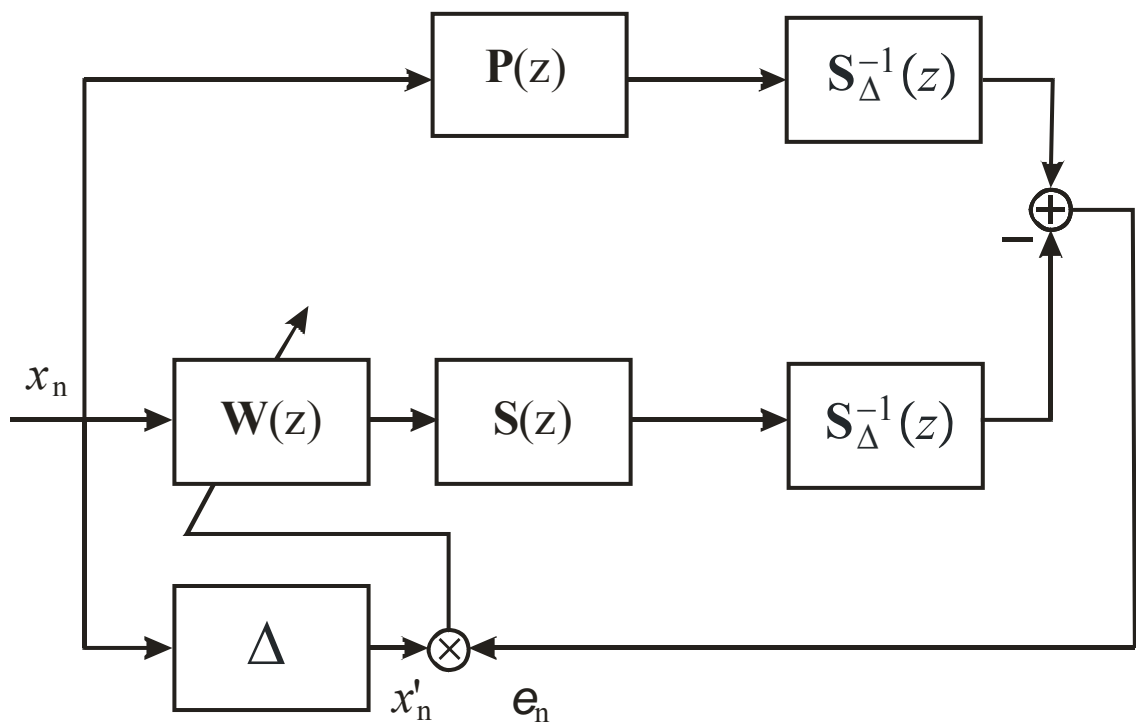


3.13. ábra. A Filtered- $\epsilon$  algoritmus működése

A 3.13. ábrán  $P(z)$  jelöli az elsődleges utat,  $S(z)$  a másodlagos utat,  $S_{\Delta}^{-1}(z)$  pedig a másodlagos átviteli út késleltetett inverzét, amit a másodlagos út mérése után off-line kell előállítani a szaggatott vonallal bekeretezett módon. A késleltetett inverz azt jelenti, hogy az adott szűrőnek, és inverzének kaszkádba kapcsolásának eredő átvitele a bemenet  $\Delta$  ütemmel való késleltetése. Ezt az eredő átvitelt, azaz a  $\Delta$  ütemnyi késleltetést jelöltem az ábrán a  $\Delta$  felíratú blokkal. Ezzel el is érkeztem az eljárás legkörülményesebb pontjára, ami  $\Delta$  megválasztása. Ezt a számot úgy kell megválasztani, hogy a modellezés után adódó inverz kellő pontosságú legyen. Ez igen nehéz, főleg akkor, ha a másodlagos átviteli út késleltetése változik, mivel akkor  $\Delta$ -t is változtatni kell, hogy a kellő pontosság megmaradjon.

### 3.3.1. Analízis

A Filtered- $\varepsilon$  LMS algoritmus analízise visszavezethető a Filtered-X LMS-re, ha a 3.13. ábrán látható blokkvázlatot átrendezzük úgy, ahogyan az a 3.14. ábrán látható.



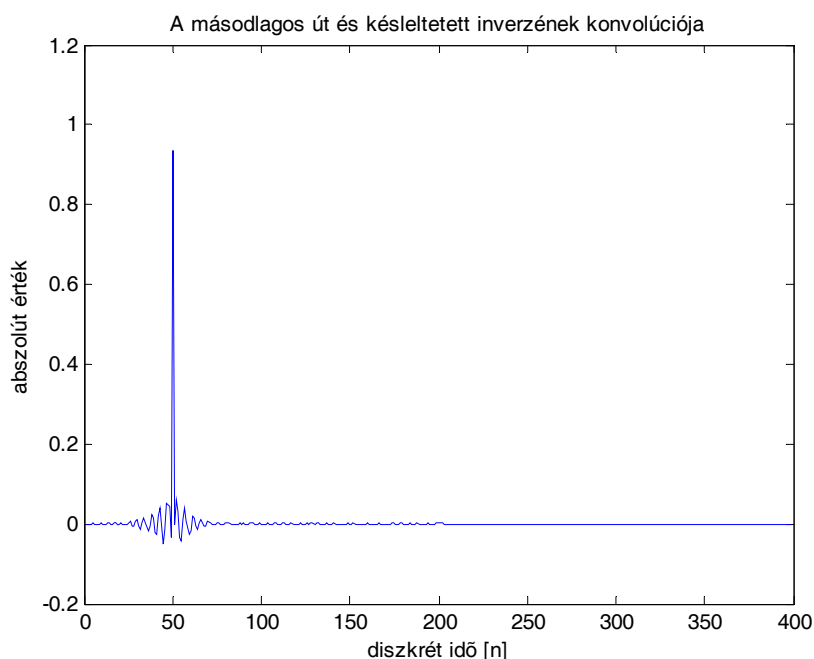
3.14. ábra. A Filtered- $\varepsilon$  LMS algoritmus analízise

A 3.14. ábra ekvivalens a 3.13-al, ugyanakkor felfedezhetjük benne magát a Filtered-X LMS algoritmust, csak itt az elsődleges út  $P(z) S_{\Delta}^{-1}(z)$ , a másodlagos út pedig  $S(z) S_{\Delta}^{-1}(z) = \Delta$ . Ha ezeket behelyettesítjük a 3.44. képletbe, akkor a következő eredményre jutunk:

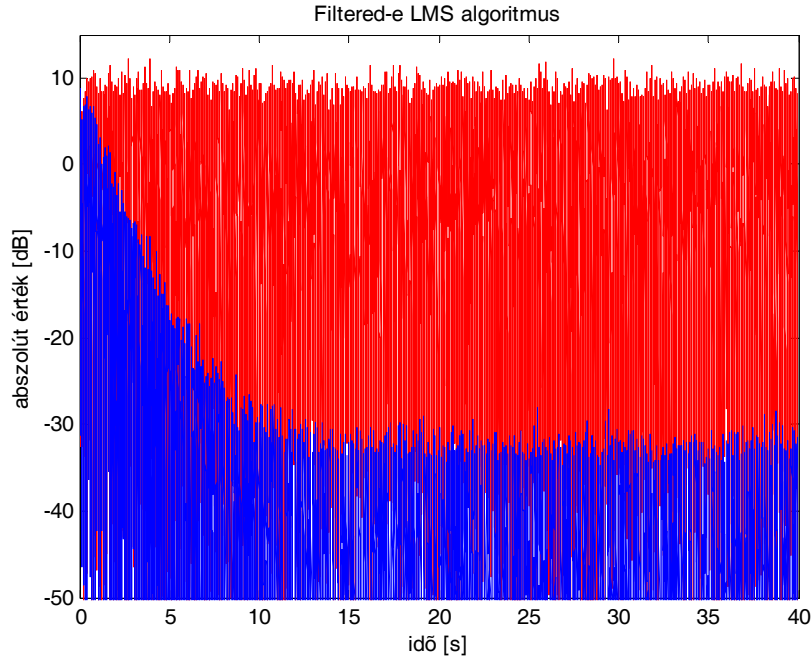
$$W(z) = \frac{P(z) S_{\Delta}^{-1}(z)}{S(z) S_{\Delta}^{-1}(z)} = \frac{P(z)}{S(z)} \quad (3.52)$$

Ami ismét a Wiener-megoldást adja. Látható tehát, hogy a Filtered- $\epsilon$  LMS algoritmus ugyanazt a végeredményt adja a  $W(z)$  adaptív szűrőre, mint a Filtered-X LMS algoritmus, a különbség ott van, hogy a Filtered- $\epsilon$  LMS algoritmussal sokkal nagyobb konvergenciasebesség érhető el, azaz gyorsabb beállítás. Így ez az eljárás képes gyorsabban követni a változásokat, mint az előző fejezetben tárgyalt Filtered-X LMS, hátránya azonban ahhoz képest, hogy kicsivel nagyobb számítási kapacitást, és bonyolultabb modellezést igényel. A nagyobb konvergenciasebesség magyarázatát a következő fejezetben mutatom be, ahol az itt bemutatott algoritmushoz hasonló, de a bonyolult inverz modellezést megkerülő Filtered-EX LMS algoritmust mutatom be.

A következőkben bemutatom a Filtered- $\epsilon$  LMS algoritmus MATLAB környezetben megvalósított szimulációs eredményeit. A szimulációs beállítások megegyeznek a Filtered-X LMS algoritmus esetében alkalmazottakkal, azaz: az elnyomandó jel normális eloszlású zérus várható értékű, egységnyi varianciájú zaj volt, az elsődleges utat egy egyszerű egységnyi átvitelű, 50 késleltetésű FIR szűrővel modelleztem, míg a másodlagos út a 3.37. egyenlettel adott IIR szűrő volt. A másodlagos út modellezését off-line végeztem egy 200 együtthatós FIR szűrővel, az egyszerű LMS algoritmust használva. Az alkalmazott  $W(z)$  szűrő szintén 200 együtthatós, a bátorsági tényező  $\mu=0.0002$  volt. A mintavételi frekvencia ebben az esetben is 2 kHz volt. A késleltetett inverz megalkotásánál a 3.13 ábra szaggatott vonallal körbevett részén megadott blokkvázlatot használtam,  $\Delta=50$  késleltetéssel. A késleltetett inverz előállítására használt bátorsági tényező szintén  $\mu=0.0002$  volt. A fenti beállításokkal a 3.15. és a 3.16. ábrán látható eredményeket kaptam:



3.15. ábra. A másodlagos út modellje és késleltetett inverzének konvolúciója



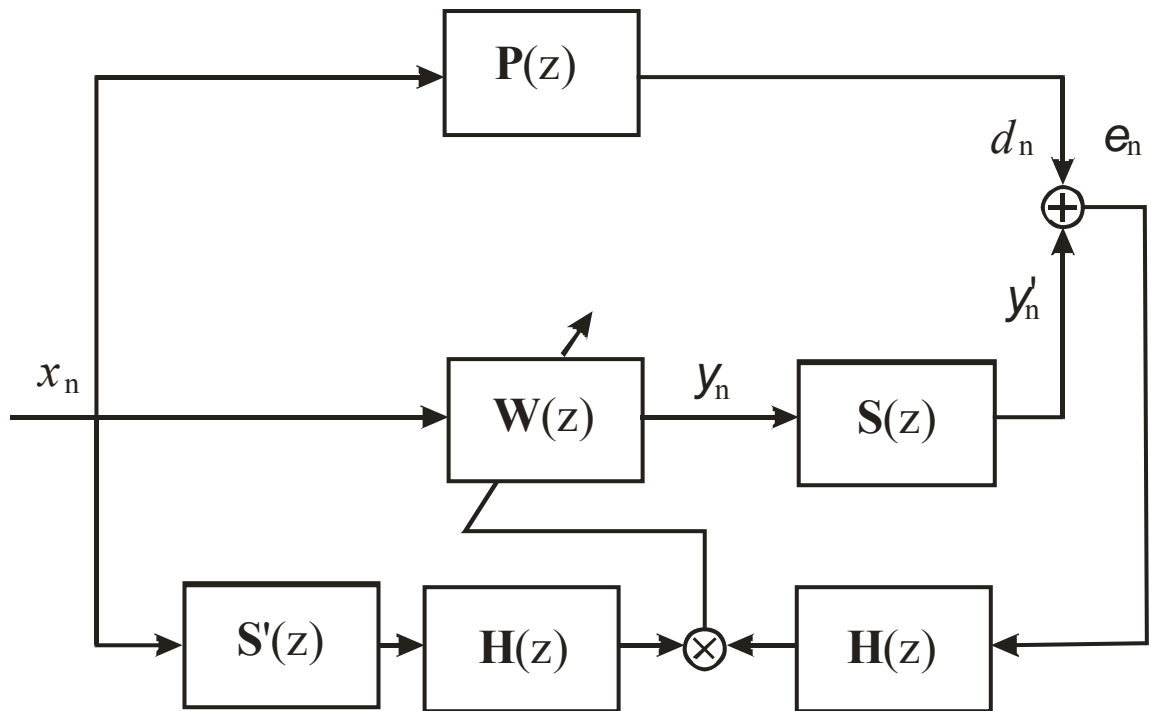
3.16. ábra. A Filtered- $\varepsilon$  algoritmus működése; mintavételi frekvencia 2 kHz

A 3.15. ábrán nagyon jól látható a késleltetett inverz, a két szűrő ( $S(z)$  és  $S^{-1}(z)$ ) kaszkádkapcsolásának eredménye egy olyan FIR szűrő, amelynek az impulzusválaszában szinte csak az 50. elem különbözik nullától, és az egységnyi, azaz a bemenetet 50 mintával késleltetve adja a kimenetére. A 3.16. ábrát nézve szembeszökő a különbség a Filtered-X LMS algoritmushoz képest, a konvergenciasebesség nagyságrendekkel nagyobb, ami azt jelenti, hogy a zajcsökkentő rendszer már 10-12 s alatt eléri az állandósult állapotot. Ez azt jelenti, hogy a Filtered- $\varepsilon$  LMS algoritmus sokkal jobban képes követni a változásokat, mint a korábban bemutatott Filtered-X LMS.

### 3.4. A Filtered-EX LMS algoritmus [6]

A Filtered-EX LMS algoritmus az előbb bemutatott Filtered- $\varepsilon$  LMS algoritmushoz hasonló módon javít a Filtered-X LMS algoritmus rossz konvergenciatulajdonságain, egy új oldaláról közelítve meg a problémát. Eszerint a Filtered-X LMS algoritmus rossz konvergenciatulajdonságai a másodlagos út amplitúdókarakterisztikájának nagy dinamikájából származnak. Ugyanis az adaptáció konvergenciasebessége az adaptációs út hurokerősítésén múlik, ami pedig arányos a másodlagos átviteli út amplitúdókarakterisztikájának négyzetének értékével, az LMS algoritmus adaptációs szabályának következtében. Az LMS algoritmus konvergenciasebességét a  $\mu$  bátorsági tényezőn keresztül szabályozzuk. Ha ennek értéke egy optimum közelében van, akkor a konvergencia gyors, ha nem, lassú. A Filtered-X LMS algoritmus esetén a bátorsági tényező maximális értékét  $S(z)$  abszolút értékének maximuma határozza meg, hiszen  $\mu$ -t olyanra kell megválasztani, hogy az algoritmus még ezen a frekvencián is stabil legyen. Ha azonban  $S(z)$  dinamikája igen nagy, akkor lesznek olyan frekvenciák, ahol a hurokerősítés nagyon kicsire adódik, ami egyben azt is jelenti, hogy az adaptáció nagyon lassú lesz.

A Filtered-EX LMS algoritmus úgy oldja a fent vázolt problémát, hogy mind a referencia, mind a hibajel megszűri, innen a név szűrt X (referenciajel) és E (hibajel). Ez a 3.17. ábrának megfelelő módon történik:



3.17. ábra. A Filtered-EX LMS algoritmus blokkvázlata

Ahol  $S'(z)$  a másodlagos átviteli út,  $S(z)$ , modellje,  $H(z)$  pedig egy FIR szűrő, ami úgy lett megtervezve, hogy  $S'(z)$ -vel és  $S(z)$ -vel kaskád kapcsolásban működve az eredő amplitúdó az egységnyi körül ingadozzon. A 3.16. ábráról jól látható, hogy a rendszer tulajdonképpen a Filtered-X LMS algoritmus módosítása. Az új elem, ami a konvergenciasebességet javítja:  $H(z)$ , ahogyan már az előbbieken említettem. Az adaptáció folyamata pedig a (3.53) és (3.54) egyenletekkel írható le:

$$\mathbf{w}_{n+1} = \mathbf{w}_n + \mu H(z) e_n \mathbf{r}_n \quad (3.53)$$

$$\mathbf{r}_n = H(z) S'(z) \mathbf{x}_n \quad (3.54)$$

### 3.4.1. Analízis

Mivel  $H(z)$  a referenciajel és a hibajel útjában jelen van, a rendszer analízise a Filtered- $\epsilon$  LMS algoritmushoz hasonlóan visszavezethető a Filtered-X LMS algoritmus analízisére, ahogyan azt a 3.14. ábrán láttuk. Ebből következik, hogy a rendszer stabil, és az adaptáció a Wiener-megoldáshoz konvergál.

A következőkben ismertetem a  $H(z)$  szűrő tervezésének fő feltételeit. Az Filtered-EX LMS struktúrában  $S'(z)$  felelős a rendszer stabilitásáért, ami azt jelenti, hogy a modellezés fázishibája nem haladhatja meg a  $\pi/2$  értéket, míg  $H(z)$  a gyors konvergenciáért felel. Ennek megfelelően:

$$|H(z)| \approx \frac{1}{|S'(z)|} \quad (3.55)$$

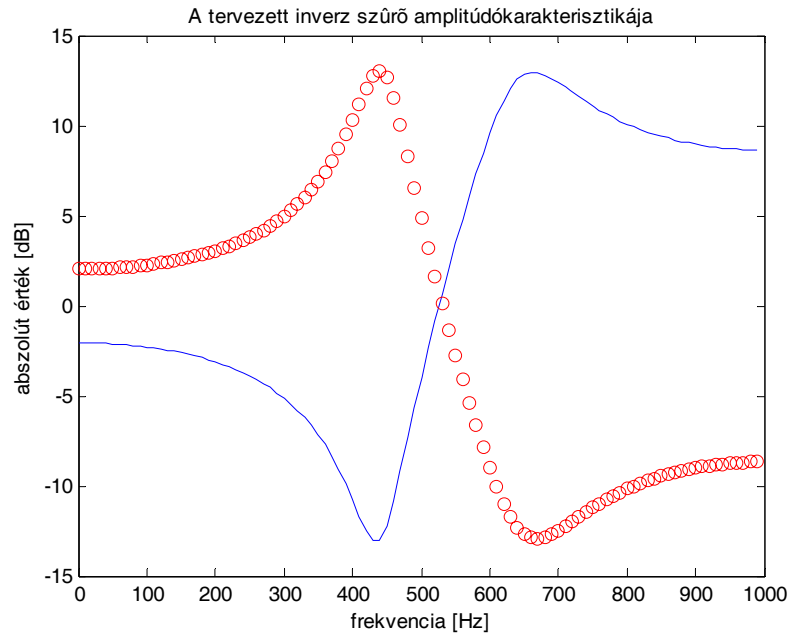
Mivel  $H(z)$  feladata csak az, hogy az  $S(z)$  nagy dinamikájából származó átvitelt egy körül ingadozó eredő átvitellé szelídítse a gyors beállást elérendő, ezért tervezésekor a modellezési hiba nagyobb lehet, mint amit a szűrőtervezésnél megszoktunk. Ez természetesen azt jelenti, hogy  $H(z)$   $S'(z)$ -hez képest már kevés együtthatóval is megvalósítható, ami csökkenti az igényelt számítási kapacitást a megvalósításkor.

A fentiek alkalmazásával a rendszer hurokerősítése ( $L(z)$ ) a következőképpen adódik, ha feltesszük, hogy  $S'(z)$  megközelítőleg egyenlő  $S(z)$ -vel:

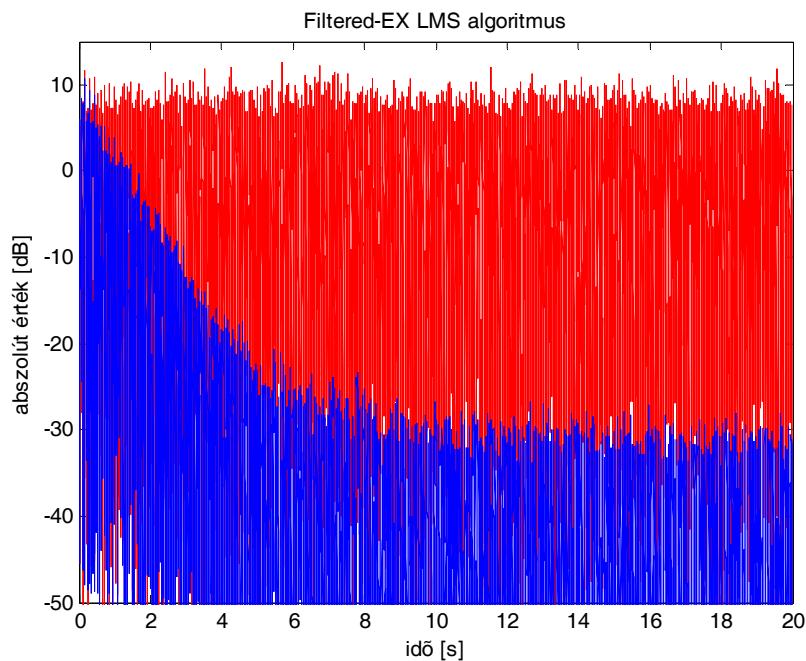
$$L(z) = \mu |H(z)S(z)|^2 \quad (3.56)$$

Aktív zajcsökkentő rendszerekben a  $\mu$  bátorsági tényezőt úgy állítjuk be, hogy a lehető legjobb konvergenciasebességet, azaz a lehető legrövidebb beállási időt érjük el. A tapasztalatok azt mutatják, hogy 6 dB ingadozás  $\mu$  értékében a konvergenciasebességben nem hoz észrevehető változásokat. Ezért  $H(z)$ -t úgy kell megterveznünk, hogy az  $S(z)$  illetve  $S'(z)$  szűrőkkel alkotott eredő átvitele az egységnyi átvitel körül megközelítőleg 3 dB-es sávban ingadozzon. Arra, hogy  $H(z)$ -t milyen eljárással tervezzük, nincs előírás, a gyakorlatban érdemes az egyszerű, és real-time környezetben is könnyen megvalósítható frekvencia mintavételező szűrőtervezési eljárást használni, mivel ez csupán egy FFT és egy IFFT lefuttatását igényli a szűrőegyütthatóknak megfelelő pontszámmal, és garantálja a megadott átviteli értékeket az adott frekvenciapontokban. A szűrőtervezésnél még ügyelni kell arra, hogy ha  $H(z)$  együtthatószámát túl nagyra választjuk, akkor a beállási időt növekedhet  $H(z)$  nagy késleltetése miatt.

A rendszer szimulációját megvalósítottam MATLAB környezetben, az alábbi beállításokkal: az elnyomandó jel normális eloszlású, zérus várható értékű, egységnyi varianciájú zaj volt, az elsődleges utat egy egyszerű egységnyi átvitelű, 50 késleltetésű FIR szűrővel modelleztem, míg a másodlagos út a (3.37) egyenlettel adott IIR szűrő volt. A másodlagos út modellezését off-line végeztem egy 200 együtthatós FIR szűrővel, az egyszerű LMS algoritmust használva. Az alkalmazott  $W(z)$  szűrő szintén 200 együtthatós, a bátorsági tényező  $\mu=0.0002$  volt. A mintavételi frekvencia ebben az esetben is 2 kHz volt. A késleltetett inverz megalkotásánál a frekvenciamintavételező szűrőtervező eljárást használtam, a tervezett  $H(z)$  szűrő 200 együtthatós volt. Az inverz szűrő és a másodlagos átviteli út modelljének amplitúdókarakterisztikája látható a 3.18. ábrán egymásra rajzolva. Pirossal jelöltem az inverz, míg késsel az eredeti szűrőt. A 3.19. ábrán az így kapott rendszer működése látható. Pirossal jelöltem az elnyomás nélküli, késsel az elnyomás bekapcsolásával elért állapotot.



3.18. ábra. A másodlagos átviteli út modellje  $S'(z)$ , és a belőle tervezett  $H(z)$ ; mintavételi frekvencia 2 kHz



3.19. ábra. A Filtered-EX LMS algoritmus működése; mintavételi frekvencia 2 kHz

Az ábráról jól látható, hogy a Filtered-EX algoritmus megközelítőleg ugyanolyan elnyomási és konvergenciaparaméterekkel rendelkezik, mint az előző fejezetben bemutatott Filtered- $\epsilon$  LMS algoritmus. A különbség a kettő között ott rejlik, hogy a Filtered-EX algoritmus által kínált konvergenciasebességet javító megoldást egyszerűbben és hatékonyabban lehet megvalósítani real-time működés közben.



## 4. Periodikus jelmodell alapú eljárások

A gyakorlatban a zajok, zavarhatások jelentős része köznyelven fogalmazva zúgó, bűgő hang, mint például egy repülő turbinájának vagy egy autó motorjának a hangja. Ez azt jelenti, hogy azoknak a zajhatásoknak a jelentős része, amelyeket aktív zajcsökkentéssel szeretnénk csillapítani, periodikus jel. Természetesen ezekre a jelekre az előző fejezetben említett szélessávú zajcsökkentő eljárások is használhatóak, de mivel a periodikus jelek leírására szolgáló apparátus, valamint az ilyen rendszerek vizsgálatához szükséges módszerek is különböznek a szélessávú zajok leírásától és vizsgálati módszereitől, ez indokolja azt, hogy egy másfajta struktúrát használjunk a periodikus jelek elnyomására [10].

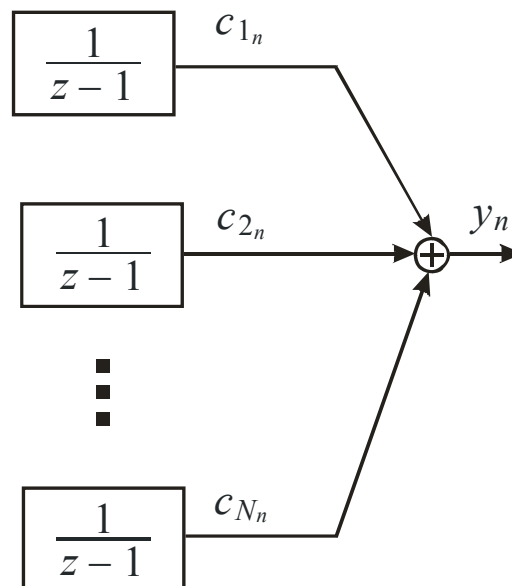
Periodikus jelek leírására a Fourier-reprezentáció használható. Ez azt jelenti, hogy egy sávkorlátozott, periodikus jel ( $p_n$ ) egyértelműen megadható a Fourier-sorával:

$$p_n = \sum_{k=-L}^L P_k c_{n_k} \quad (4.1)$$

$$c_{n_k} = e^{j2\pi fkn}; \quad k = -K \dots K \quad (4.2)$$

A fenti két képletben  $f$  a periodikus jel alapharmonikusának frekvenciája,  $k$  véges sok értéke pedig a sávkorlátozottságot fejezi ki.

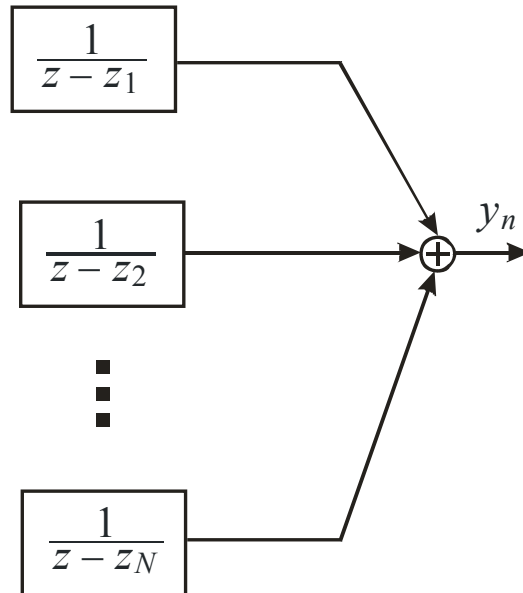
A (4.1) és (4.2) képletek segítségével megalkothatjuk a periodikus jelek koncepcionális jelmodelljét, azaz meghatározhatunk egy olyan struktúrát, amely periodikus jelek generálására képes. Ez a modell látható a 4.1. ábrán:



4.1. ábra. Sávkorlátozott periodikus jel modellje

A fenti ábra alapján egy  $p_n$  sávkorlátozott periodikus jel úgy jön létre, hogy az integrátorok bemenetére a nulla időpillanatban a (4.1) egyenletnek megfelelő  $P_k$  Fourier-

együtthatókat adjuk, majd a továbbiakban nullát adunk az integrátorok bemenetére. Ennek következtében az integrátorok tartani fogják a nulla időpillanatban felvett értékeiket, így a modell pontosan a (4.1) egyenlet által megfogalmazottakat valósítja meg. Sávkorlátozott jelek generálására egy másik, az előzővel ekvivalens modell is használható, ezt mutatja a 4.2. ábra:

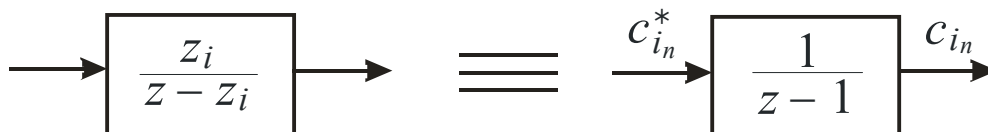


4.2. ábra. Sávkorlátozott periodikus jel modellje

A fenti ábrán  $z_i$  a következőképpen határozható meg:

$$z_i = \frac{c_{n+1_i}}{c_{n_i}} \quad (4.3)$$

A rezonátoros és az integrátoros modell megfeleltetését a következő ábra mutatja:

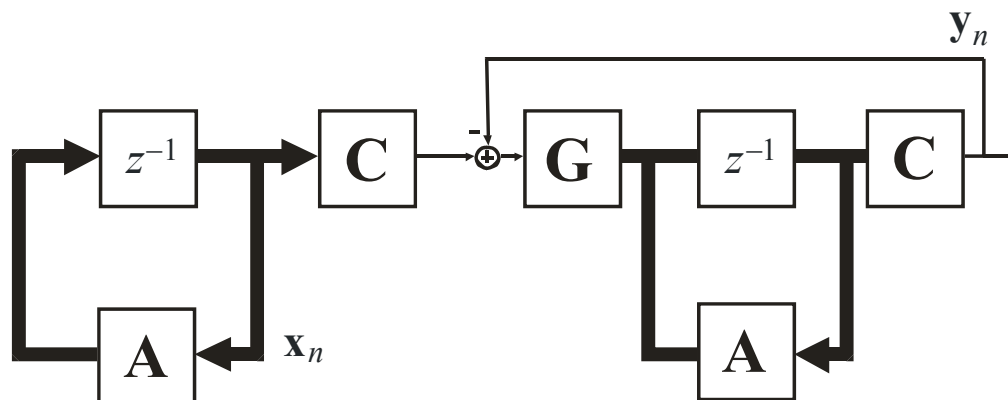


4.3. ábra. A rezonátoros és az integrátoros jelmodell megfeleltetése

A következő fejezetben megmutatom, hogy a fenti jelmodellekhez hogyan tervezhető megfigyelő, és ez a struktúra miként használható zajcsökkentési feladatokra.

#### 4.1. Jelmodell alapú megfigyelő, a rezonátoros struktúra

A megfigyelők olyan rendszerek, amelyek egy másik rendszer állapotváltozói, vagy az azokból származtatott mennyiségek meghatározását teszik lehetővé. Tehát olyan mérési eljárást valósítanak meg, melynek során egy rendszer mérés technikailag nem hozzáférhető állapotváltozóinak lemásolásával azok mérését teszik lehetővé. A mérendő rendszer és a megfigyelő kapcsolatát mutatja a 4.4. ábra [2]:



4.4. ábra. A megfigyelő és a megfigyelt rendszer viszonya

Az ábrán a megfigyelt rendszer állapotváltozóit  $\mathbf{x}$ -szel jelöltük, az állapotátmeneti mátrixot pedig  $\mathbf{A}$ -val. A  $\mathbf{C}$ -vel jelölt mátrixot kicsatoló mátrixnak nevezzük. Ezt a bal oldalon álló független rendszert figyeljük meg az ábra jobb oldalán található megfigyelővel.

A megfigyelőt úgy kell megalkotni, hogy fel kell állítanunk a mérendő rendszer egy koncepcionális jelmodelljét, ami esetünkben már megtörtént, majd az  $\mathbf{A}$  és  $\mathbf{C}$  mátrixok segítségével még egyszer meg kell valósítani a mérendő rendszert, kiegészítve az egy bemenettel. Ez a bemenet fogja szolgáltatni a mérendő rendszer és az általunk megépített kópia kimenete közti különbséget, amit egy  $\mathbf{G}$  becsatoló mátrixon keresztül vezetve arra használunk, hogy az általunk megépített rendszer állapotváltozóit módosítsuk oly módon, hogy a valós rendszer és a kópia kimenete közötti különbség minél kisebb legyen.

A megfigyelt rendszer állapotváltozós leírása tehát:

$$\mathbf{x}_{n+1} = \mathbf{A}\mathbf{x}_n \quad (4.4)$$

$$\mathbf{y}_n = \mathbf{C}\mathbf{x}_n \quad (4.5)$$

Most felírjuk a megfigyelő állapotváltozós leírását, a megfigyelő állapotváltozóit  $\mathbf{x}'$ -vel jelölve:

$$\mathbf{x}'_{n+1} = \mathbf{A}\mathbf{x}'_n + \mathbf{G}\{\mathbf{y}_n - \mathbf{y}'_n\} \quad (4.6)$$

$$\mathbf{y}'_n = \mathbf{C}\mathbf{x}'_n \quad (4.7)$$

Ha a (4.6)-ba visszahelyettesítjük (4.4)-et és (4.5)-öt, akkor algebrai átalakítással belátható, hogy a megfigyelő állapot átmeneti mátrixa  $\mathbf{A}-\mathbf{G}\mathbf{C}$ . A megfigyelő akkor működik

megfelelően, ha a kópia és a megfigyelt rendszer állapotváltozói közti különbség fokozatosan csökken, és előbb-utóbb nullává válik. A hibára a következő egyenlet írható fel:

$$\mathbf{x}_{n+1} - \mathbf{x}'_{n+1} = \{[\mathbf{A} - \mathbf{GC}]^{n+1} (\mathbf{x}_0 - \mathbf{x}'_0)\} \quad (4.8)$$

A fenti képletből látható, hogy a hiba akkor lesz nulla, ha  $\mathbf{A} - \mathbf{GC}$  már eleve nulla, ez abban az esetben lehetséges, ha a  $\mathbf{C}$  négyzetes, azaz annyi kimenete van a megfigyelt rendszernek, ahány állapotváltozója, és még az is szükséges, hogy ezekből a kimeneteből  $\mathbf{C}$  állapotváltozóira vissza lehessen következtetni. Ez tehát azt jelenti, hogy  $\mathbf{C}$ -nek négyzetesnek kell lennie, és sor illetve oszlopvektorainak függetleneknek. Ha a fenti feltétel teljesül, akkor egy lépéses konvergencia biztosítható akkor, ha  $\mathbf{G} = \mathbf{AC}^{-1}$  választással élünk. Ha  $\mathbf{C}$  nem négyzetes, akkor is nullává válhat a hiba, ha valamely  $n$ -re  $(\mathbf{A} - \mathbf{GC})^n = 0$  adódik. Ekkor az  $(\mathbf{A} - \mathbf{GC})$  mátrix nilpotens. A nilpotens mátrixokra pedig az jellemző, hogy mindegyik sajátértékük nulla. Ez egyben azt is jelenti, hogy a rendszer pólusai az origóban helyezkednek el, vagyis a rendszerünk FIR. Ez az aktív zajcsökkentő rendszerek szempontjából is előnyös, mivel ha a megfigyelő véges lépésben beáll, akkor feltételezhető, hogy a megfigyelt rendszert igen gyorsan tudja követni.

A fentiek ismeretében a feladatunk már csupán annyi, hogy a periodikus jelek koncepcionális modelljét használva felírjuk parametrikusan az  $(\mathbf{A} - \mathbf{GC})$  mátrixot, és kihasználva azt a feltételt, hogy mindegyik sajátértéknek nullának kell lennie, megoldjuk  $\mathbf{G}$  elemeire az egyenletrendszert. Tehát a rezonátoros jelmodellre:

$$\mathbf{A} = \langle z_i \rangle \quad (4.9)$$

$$\mathbf{C} = \mathbf{c}^T = [111\dots 1] \quad (4.10)$$

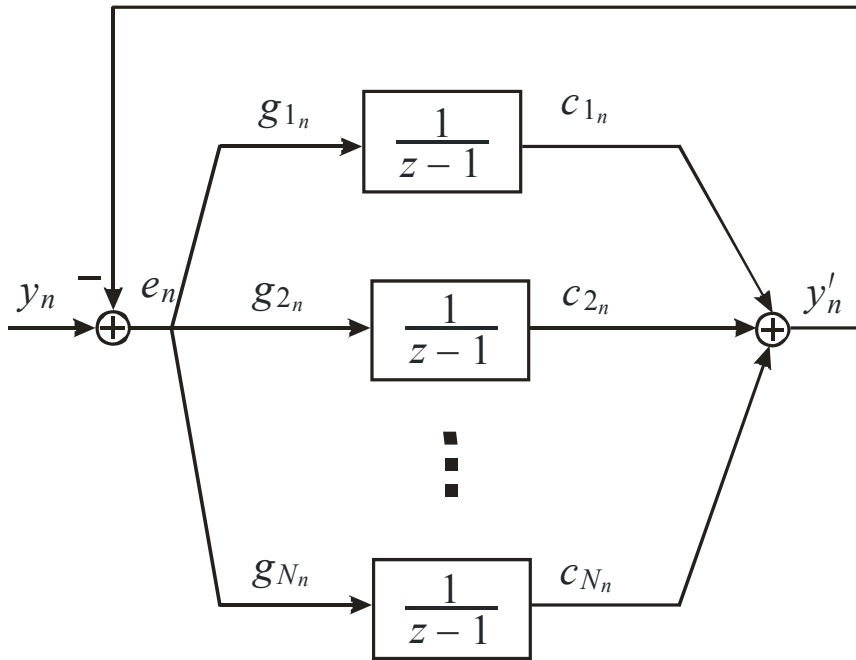
Az egyenleteket a fenti gondolatmenet szerint megoldva, véges beállású megfigyelőt megcélözva, valamint azt, hogy a rezonátorpólusok az egységkörön egyenletesen helyezkednek el, az eredmény:

$$g_i = z_i \frac{1}{N} \quad (4.11)$$

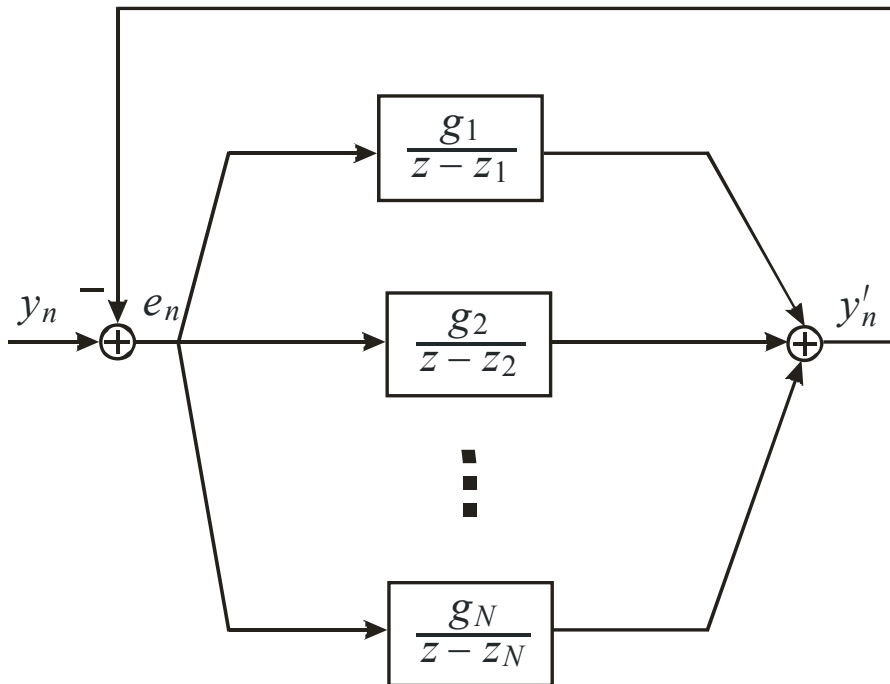
A fenti egyenletben  $N$  a rezonátorok száma,  $z_i$  az  $i$ -edik rezonátorhoz tartozó pólus. Természetesen az integrátoros jelmodellhez is tervezhető megfigyelő. Erre az eredmény az előző esetben használt feltételeket megtartva:

$$g_{ni} = c_{ni}^* \frac{1}{N} \quad (4.12)$$

A fenti képletben a csillag a konjugálást jelöli,  $n$  a diszkrét időt jelöli, míg  $i$  azt, hogy hányadik rezonátorról van szó,  $N$  pedig a rezonátorszám. A jelmodellekhez tartozó megfigyelőket mutatja a 4.5. és a 4.6. ábra [12]:

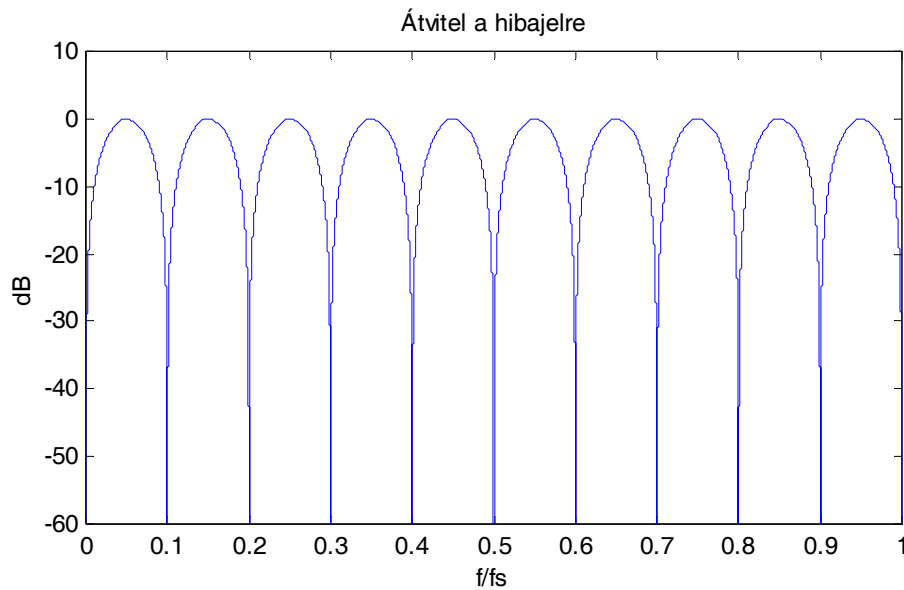


4.5. ábra. Rezonátoros megfigyelő integrátorokkal megvalósítva

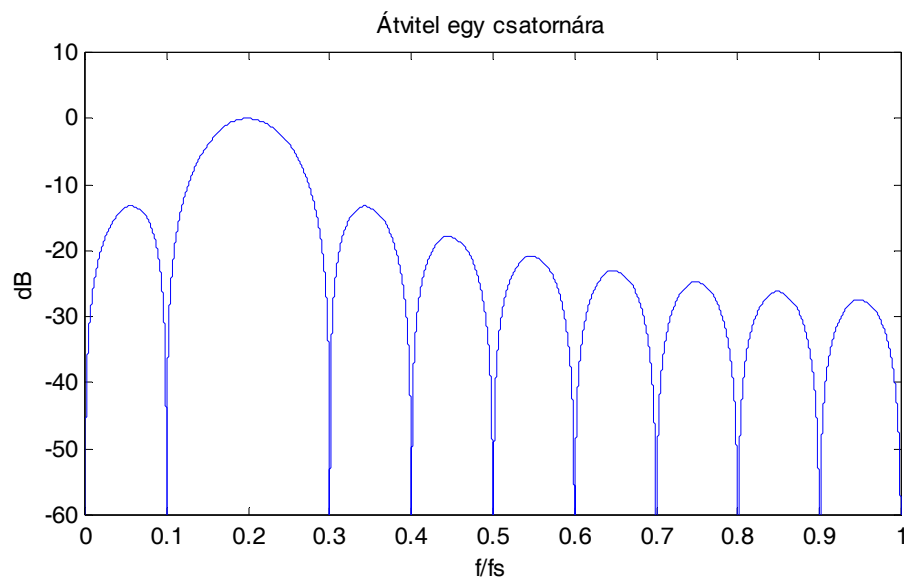


4.6. ábra. Rezonátoros megfigyelő rezonátorokkal megvalósítva

A rezonátoros struktúra ismertetését néhány jellemző átviteli függvény megadásával zárjuk. A következő két ábra közül az első a hibajel (a 4.5. és 4.6. ábrákon  $e_n$  jelöli) amplitúdókarakterisztikáját mutatja, a második az  $i$ -edik csatorna (egy csatorna az egy integrátoron keresztmenő jelfolyam) átvitelét mutatja  $N=10$  esetén.



4.7. ábra. Rezonátoros struktúra átvitele a hibajelre a relatív frekvencia függvényében



4.8. ábra. Rezonátoros struktúra átvitele egy csatornára a relatív frekvencia függvényében

Az ábrákból jól látható, hogy a rezonátoros struktúra azokon a frekvenciákon, ahol rezonátorpólusok vannak, nulla hibával előállítja a jelet, más frekvenciákon a hiba valamilyen véges érték. A második ábra, ami egy csatorna átvitelét mutatja, jól mutatja, hogy egy csatorna csak a saját frekvenciáján biztosít egységnyi átvitelt, sőt, a többi rezonátorfrekvencián az átvitele nulla.

A fentiekben bemutatott rezonátoros struktúra a gyakorlatban is sikeresen alkalmazható, saját alkalmazásaimban az integrátoros megfigyelőt használtam egyszerűbb megvalósíthatósága miatt.

## 4.2. Adaptív Fourier-analízis

Az előző alfejezetben láttuk, hogy a rezonátoros struktúra a rezonátorfrekvenciákon pontosan előállítja a bemenő jelet. Gyakorlati alkalmazásokban azonban az elnyomni kívánt jel frekvenciája változhat. Ilyen eset például, ha egy autó motorjának túlságosan zúgó hangját szeretnénk elnyomni; ha az autó gyorsul, vagy fékezik, az elnyomandó jel frekvenciája változik. Mivel az elnyomást ilyen esetekben is biztosítani kell, a rezonátorpólusokat az elnyomandó jel frekvenciájának megfelelően hangolni kell. Ezt a feladatot látja el az Adaptív Fourier Analizátor (AFA).

Az AFA a rezonátorok pólusait az elnyomandó jel frekvenciájára hangolja, ezzel biztosítja a rezonátoros struktúra számára, hogy az elnyomandó jel változásait követni tudja, így a zajelnyomásra a nulla hiba megcélozható. Ehhez természetesen szükség van a zajforrásból származó referenciajelre. Ez általában nem okoz problémát, valamilyen egyszerű szenzor alkalmazásával könnyen megoldható.

Az AFA tulajdonképpen egy rezonátoros struktúra, működését legszemléletesebben a rezonátoros struktúra integrátorokkal való megvalósítását mutató ábra segítségével lehet bemutatni. A jelmodell ebben az esetben:

$$\mathbf{y}_n = \mathbf{c}_n^T \mathbf{x}_n \quad (4.13)$$

$$c_{n,k} = e^{j2\pi \frac{1}{N} kn}; \quad k = -K \dots K; \quad N = 2K + 1 \quad (4.14)$$

A fenti esetben az alapharmonikus relatív frekvenciája  $1/N$ . Az állapotváltozós leírás a következőképpen írható fel:

$$\mathbf{x}'_{n+1} = \mathbf{x}'_n + \frac{1}{N} \mathbf{c}_n^* \{ \mathbf{y}_n - \mathbf{c}_n^T \mathbf{x}'_n \} \quad (4.15)$$

$$\mathbf{y}'_n = \mathbf{c}_n^T \mathbf{x}'_n \quad (4.16)$$

Egy csatorna működése úgy magyarázható meg szemléletesen, hogy a hibajelel először a  $c_{n,k}^*$  együttható DC-re keveri, majd integrálás után a megfelelő  $c_{n,k}$  együttható keveri vissza a jelet az eredeti  $f_k$  frekvenciára. Ha a megfigyelő pontosan illeszkedik a jel frekvenciájához, akkor az állapotváltozó nem változik; ha ez nem teljesül, akkor az állapotváltozó egy forgó komplex vektor lesz, a forgás sebessége pedig az aktuális frekvenciakülönbségnek felel meg. Ezért ezt (az állapotváltozó két időpillanat közötti szögeltérését) használjuk fel a frekvencia adaptálására:

$$f_{1_{n+1}} = f_{1_n} + \frac{1}{2\pi N} \text{phase} \{ x'_{1_{n+1}} - x'_{1_n} \} \quad (4.17)$$

$$c_{k_{n+1}} = c_{k_n} e^{j2\pi f_1 (n+1)k} \quad (4.18)$$

Az előbbi két képlet azt fejezi ki, hogy a rendszer alapfrekvenciájának adaptálása után a  $c$  együtthatók 4.18. szerinti frissítésével a rendszer frekvenciája adaptálódott. Fontos megemlítenünk, hogy ha a rendszer frekvenciája nő, akkor a legnagyobb frekvencián lévő rezonátorok közelíthetnek az  $f = 0,5$  relatív frekvenciához, azaz a mintavételi frekvencia feléhez. A rendszer stabilitása és a véges beállítás szempontjából nagyon fontos, hogy a rezonátorok egymástól körülbelül egyenletes távolságra helyezkedjenek el. Ha azonban a frekvenciaadaptálás során az egyik rezonátor nagyon megközelíti az  $f = 0,5$  frekvenciát, akkor az adott frekvencián és a tükörfrekvencián lévő rezonátor egymáshoz nagyon közel kerülnek. Ez akár a rendszer instabilitását is okozhatja. Ezért, ha valamelyik rezonátor az  $f = 0,5$  frekvenciát megközelíti, akkor azt ki kell léptetni.

Hasonlóképpen, ha az alapharmonikus frekvenciája csökken, és újabb egész számszorosa még elfér az  $f = 0,5$  frekvenciáig, akkor az egyenletes rezonátor-elhelyezkedést biztosítandó új rezonátorokat kell beléptetnünk a rendszerbe. Ezeknek a rezonátoroknak az inicializálása a következő:

$$x_{n+1_K} = x_{n+1_{-K}} = 0 \quad (4.19)$$

$$c_{n+1_K} = c_{n+1_{-K}} = 1 \quad (4.20)$$

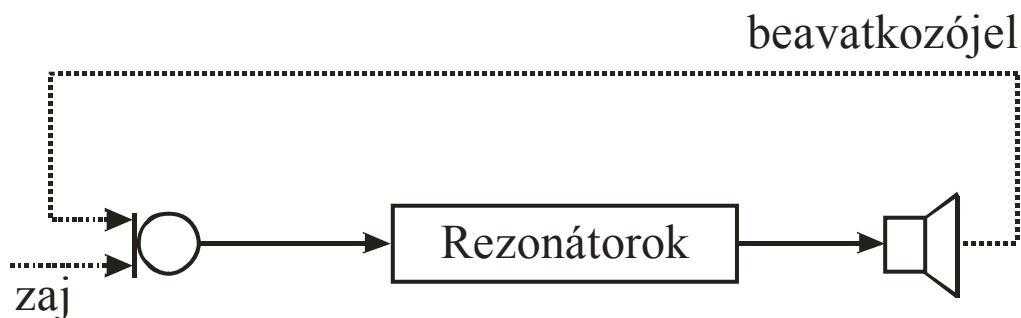
A fentiekben az AFA általam és a gyakorlatban is használt formáját ismertettem, amellyel kiegészülve a rezonátoros struktúra képes követni az elnyomandó jel alapharmonikusának változásait.

### 4.3. *Rezonátoros zajcsökkentő eljárások elemzése*

Az előző fejezetben ismertetett, AFA-val kiegészített rezonátoros struktúra nagyon jól alkalmazható olyan zajelnyomási feladatokra, melyeknél az elnyomandó jel periodikus. A rendszer előnye, hogy kis számítási igényt támaszt – a memóriában a legnagyobb helyet az exponenciálisok kiszámításához szükséges szinusztáblázat foglalja – valamint az elnyomandó jelet igen pontosan elő tudja állítani.

Ahogy az előző fejezetben láttuk, ha a frekvencia-adaptálás pontos, azaz a bemenő jel komponenseinek frekvenciái megegyeznek a rezonátorfrekvenciákkal, akkor a jel előállítás pontos, az elnyomás teljes lehet. Ha azonban a rezonátoros struktúrát összevetjük az aktív zajcsökkentő rendszer blokkvázlatát bemutató 2.2-es ábrával, akkor láthatjuk, hogy a rezonátoros struktúra esetén  $A(z)$  azonosan egyenlő eggyel. A valóságban a gyakorlati alkalmazások során azonban  $A(z)$  nem azonosan egyenlő eggyel, hanem frekvenciafüggő átvitele van. Egy ilyen zajcsökkentő rendszer blokkvázlatát mutatja az alábbi ábra.





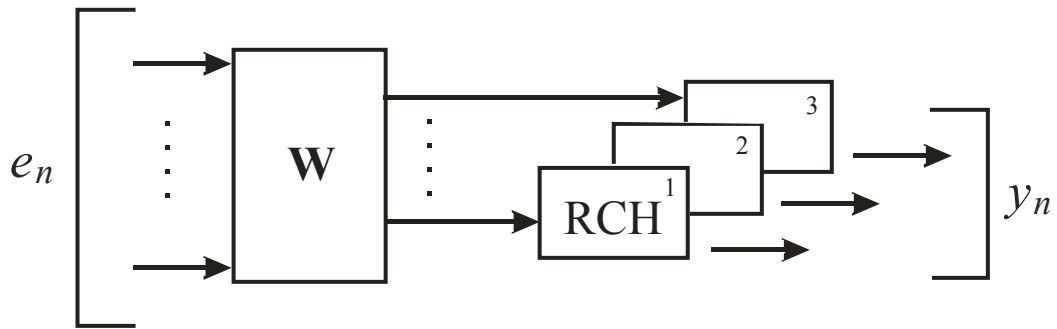
4.9. ábra. Egycsatornás zajcsökkentő rendszer rezonátoros struktúrával

Az ábra alapján jól látható, hogy az aktív zajcsökkentés esetén a megfigyelő  $A(z)$ -val szűrt kimenő jelének kell egyeznie a bemenő jellel. Ez azt jelenti, hogy a rezonátorok visszacsatolása egy külső hurkon keresztül valósul meg. A rezonátoros struktúra működéséhez azonban el kell érni, hogy a fenti feltétel ( $A(z)=1$ ) teljesüljön. Ennek érdekében szükség van  $A(z)$  inverzére, hiszen ezzel szűrve a rezonátoros struktúra kimenetét az mindenképpen keresztül megy  $A(z)$ -n, így az eredmény az lesz, hogy a késleltetéstől eltekintve biztosítottuk a  $A(z) = 1$  feltételt. A inverzének meghatározásáról a későbbiekben ejtek szót.

A maximális konvergenciasebesség eléréséhez véges impulzusválaszú (FIR) rendszer tervezését tűztem ki célul, mivel feltételeztem, hogy a FIR rendszerek beállása gyorsabb, mint az IIR rendszereké. Az ilyen rendszerek hátránya, hogy a tranziensek során nagyok a túllövésiek, állandósult állapotban pedig nagyon érzékenyek a zajokra. A túllövésiek azért nagyon veszélyesek, mivel túlsordulást okozhatnak. Akusztikus esetben azonban ez nem reális veszély.

A zajelnyomás csak akkor lehet teljes, ha, mint már említettem, a rezonátorokat AFA segítségével a megfelelő frekvenciákra hangoljuk. Ez a hangolás történhet kétféleképpen is: az első megoldás, hogy a zajcsökkentő eljárásban adaptív rezonátorkészletet alkalmazok, ekkor azonban a teljes zajelnyomó hurok felel meg az AFA-nak, így a frekvenciaadaptáció lassabb. A másik megoldás, hogy a zajforrásból vett referencijel segítségével, valamint egy külön rezonátoros struktúrával működtetem az AFA-t, a zajelnyomást végző rezonátorokat pedig a  $c_{n,k}$  együtthatók változtatásával adaptálom. Ez a két megoldás hasonló a már korábban említett előre és visszacsatolt struktúrához, azok előnyeit és hátrányait is hordozzák. Az előrecsatolt rendszer mellett döntöttem, azaz a zajforrásból vett referencijellel végeztem a frekvencia adaptálását.

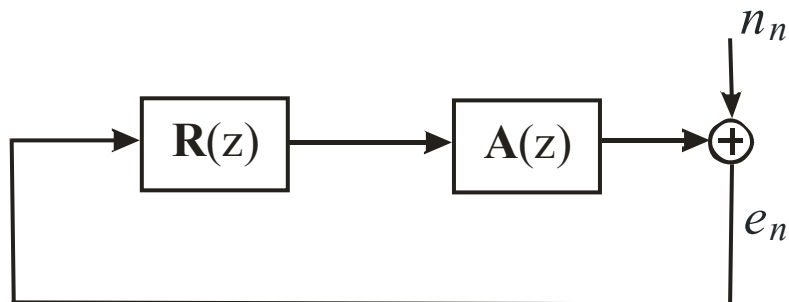
Az egycsatornás rendszerek mintájára megszerkeszthető a többcsatornás rendszer blokkvázlata is. Ebben az esetben minden kimenethez tartozik egy független rezonátoros struktúra. A frekvencia adaptálására továbbra is használható az AFA. A hibajel azonban ebben az esetben egy vektor, ezt egy  $\mathbf{W}(z)$  becsatolómátrix segítségével vezetjük a rezonátorok bemenetére. Ezt mutatja a 4.10-es ábra.



4.10. ábra. Többcsatornás zajcsökkentő rendszer blokkvázlata

### 4.3.1. Stabilitás

Ebben a fejezetben ismertetem a rezonátoros struktúra stabilitásának szükséges feltételeit egy- és többcsatornás esetben. A zajcsökkentő hurok stabilitásának vizsgálatához tekintsük az alábbi ábrát:



4.11. ábra. Zajcsökkentő rendszer blokkvázlata

Az ábrán  $R(z)$  jelöli a visszacsatolatlan rezonátoros struktúrát,  $A(z)$  pedig az akusztikus átvitelt,  $n_n$  az elnyomandó zaj,  $e_n$  a hibajel.  $R(z)$  tulajdonképpen az egy csatornára vonatkozó átvitelek összege, azaz (egy csatornás esetben):

$$R(z) = \sum_{i=1}^N Q_i(z) = \sum_{i=1}^N \frac{r_i z_i}{z - z_i} \quad (4.21)$$

A következőkben azt mutatom meg, hogy a rendszer milyen  $r_i$  választással lesz stabil, hiszen stabil rendszer és a rezonátorok megfelelő frekvenciára való hangolása esetén az elnyomás teljes. Tekintsük  $r_i$ -t a következő formában:

$$r_i = \alpha w_i \quad (4.22)$$

Egyrészt kínálkozik, hogy a hurokerősítést a szabályozási körökhöz hasonló módon kellően kicsire választva a rendszer stabil lehet. Ez  $\alpha$  kicsire választását jelentené. Erről azonban belátható, hogy a rezonátoros struktúra esetén nem alkalmazható kizárólagosan, szükség van egy további feltételre [12].

A stabilitás vizsgálatára a Nyquist-kritériumot alkalmazzuk. Ha a rezonátorokat az egységkör belsejére soroljuk, akkor a Nyquist-kritérium szerint a stabilitás feltétele az, hogy a nyílt hurok Nyquist-görbéje a  $-1$  pontot ne ölelje körül. A stabilitás vizsgálatához először a (4.21) egyenlet jobb oldalán található összeget tagonként kell vizsgálni. Így egy rezonátorra meghatározható annak Nyquist-görbéje. A Nyquist-görbe vizsgálatát elvégezve  $w_i$  és  $\alpha$  függvényében belátható, hogy egyetlen visszacsatolt rezonátor  $\alpha$  segítségével csak akkor stabilizálható, ha  $w_i$ -re teljesül, hogy:

$$|\text{arc}(w_i)| < \frac{\pi}{2} \quad (4.23)$$

Ezek után elvégezhető a teljes hurok stabilitási vizsgálata, melyben megmutatható, hogy egy adott rezonátorfrekvencia környezetében a rendszer fázisát döntően  $A(\omega)$  határozza meg. Ezek után a teljes hurok stabilitásáról belátható, hogy stabilitásának szükséges feltétele az, hogy:

$$|\text{arc}(A(z_i)) + \text{arc}(w_i)| < \frac{\pi}{2} \quad (4.24)$$

Összefoglalva:  $w_i$ -k fenti kritériumnak megfelelő megválasztásával, valamint  $\alpha$  konvergencia paraméter kellően kicsivé választásával a rendszer stabilitása biztosítható. Gyakorlati alkalmazásokban célszerű a rezonátor csatornákon a következő választással élni:

$$w_i = \frac{1}{A(z_i)} \quad (4.25)$$

Többcsatornás rendszer esetén a nyílt hurkú rendszer átviteli függvénye a következő:

$$\mathbf{F}(z) = \mathbf{A}(z) \sum_{i=1}^N \mathbf{R}_i(z) Q_i'(z) \quad (4.26)$$

ahol  $\mathbf{R}_i$  a hibajelvektort a rezonátorok bemenetére csatoló mátrix. A rezonátorokat továbbra is az egységkör belsejéhez sorolva a többdimenziós rendszerek stabilitására vonatkozó Nyquist-kritérium szerint a visszacsatolt rendszer akkor stabil, ha  $\mathbf{F}$  sajátértékei (amelyek frekvenciafüggők) nem ölelik körül a  $-1$  pontot.  $\mathbf{R}_i$ -t a következő alakban keressük:

$$\mathbf{R}_i = \alpha \mathbf{W}_i \quad (4.27)$$

A levezetésből azt kapjuk, hogy a többcsatornás rendszer stabilitásának szükséges feltétele:

$$|\text{arc}(\lambda_j)| < \frac{\pi}{2} \quad (4.28)$$

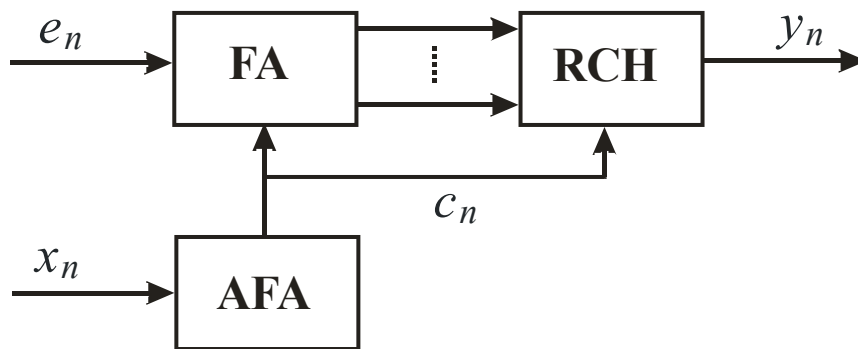
ahol az  $A(z_i)$   $W_i$  mátrix sajátértékeit  $\lambda$ -val jelöltük. Ebből következik, hogy gyakorlati alkalmazásokban a következő paraméterválasztással érdemes élni:

$$W_i = A^\#(z_i) \quad (4.29)$$

ahol # a pszeudoinverz-képzést jelöli.

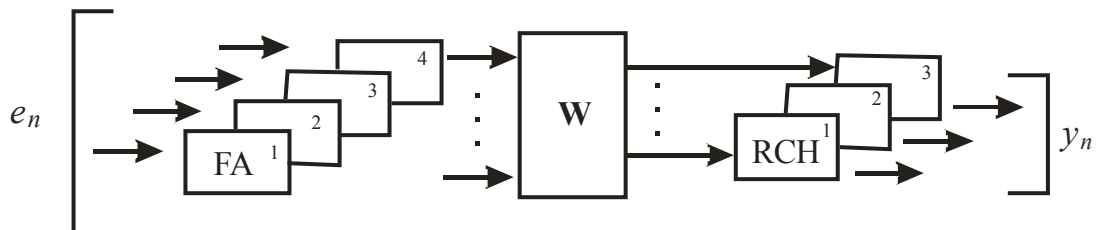
### 4.3.2. Konvergenciasebesség növelése

Ebben a fejezetben bemutatom, hogy hogyan növelhető a rezonátoros zajelnyomó rendszer konvergenciasebessége. Egy ilyen megnövelt sebességű egysatornás rendszer blokkvázlata látható a 4.12. ábrán:



4.12.. ábra. Megnövelt sebességű egysatornás zajcsökkentő rendszer blokkvázlata

Az ábrán látható rendszer abban különbözik az eddig tárgyalttól, hogy a bemenetre érkező hibajeleket az FA-val jelölt, Fourier analízátor komponenseire bontja, majd ezek a komponensek a zajcsökkentést végző RCH rezonátorok bemeneteire kerülnek. Mindkét rezonátoros struktúra rezonátorpozícióit AFA-val állítjuk. Ez az elrendezés általánosítható többsatornás esetre is, ahogy a 4.13. ábra mutatja:



4.13. ábra. Megnövelt sebességű többsatornás zajcsökkentő rendszer blokkvázlata

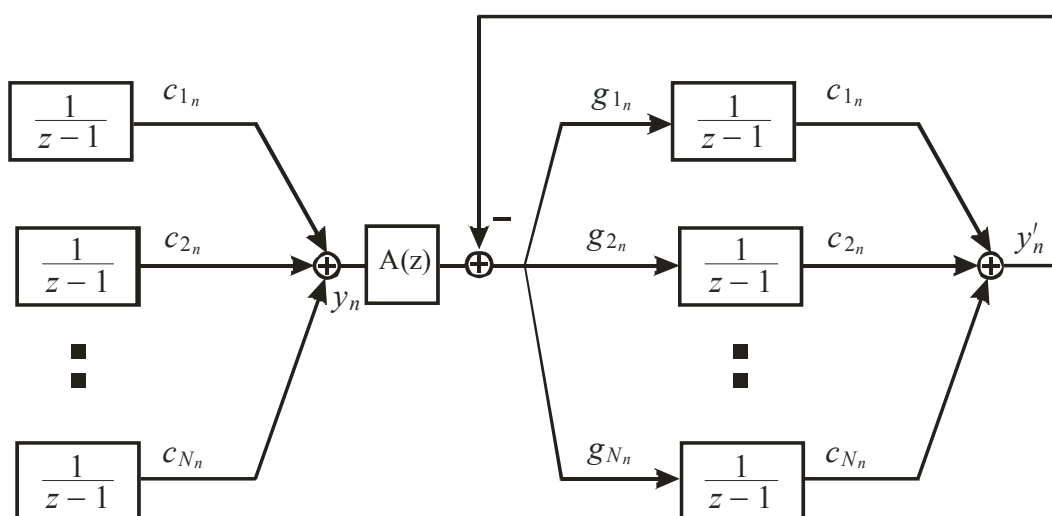
Az ábrán látható, hogy minden hibajelhez, azaz minden mikrofonhoz tartozik egy analizátorkészlet, és minden kimenethez egy rezonátorkészlet. Ez a gyakorlatban is nagyon jól használható struktúra tehát három rezonátorkészlet működtetését igényli. Igaz ugyan, hogy ez kicsit megnöveli a számítási igényt, de mindamelllett jelentős sebességnövekedést jelent.

### 4.3.3. Az akusztikus átvitel mérése [11]

Az átviteli függvény mérése általános feladat, amelyre sokféle megoldás kínálkozik, ahogyan azt a 3.1.1. fejezetben már említettem:

- gerjesztés fehér zajjal, kiértékelés FFT-vel
- gerjesztés multiszínuszos jellel, kiértékelés FFT-vel
- adaptív szűrő paramétereinek beállítása (például LMS algoritmussal).

A fenti eljárások előnyeit és hátrányait már az 1.1-es fejezetben bemutattam, ezért erre nem térek ki újra. Mindezek ellenére én egy negyedik, rezonátoros struktúra alapú átviteli függvény mérést alkalmaztam. Azért választottam ezt a struktúrát, mert zajelnyomásra rezonátoros struktúrát használok, ezért maga a rezonátoros struktúra már adott, így egy nagyon kis módosítással megkapható belőle az átviteli függvény mérésére alkalmas elrendezés, amely már korábban a tanszéken kifejlesztésre került [11]. Ezen kívül a struktúra keskeny sávú (színuszos) gerjesztést alkalmaz, ami nagyon pontos mérést tesz lehetővé, és az átlagolás is könnyen megvalósítható. Ennek a mérésnek a blokkvázlatát a következő ábra szemlélteti [11]:



4.14.. ábra. Átviteli függvény mérése rezonátoros struktúrával

A méréshez az integrátoros jelmodellt használtam, mivel ekkor állandósult állapotban az állapotváltozók maguk a Fourier-együtthatók. A bemutatott struktúrával az a probléma, hogy egyszerre csak  $N$  frekvencián tudja megmérni az átvitelt. Ennek a hibának a kiküszöbölésére a következő megoldás alkalmazható. Egyetlen AC rezonátorpárt kell működtetni, s az akusztikus rendszert egy sweepelő szinuszzel kell gerjeszteni. A szinusznak annyi frekvencián kell végmennie, ahány pontban az átvitelt meghatározni szeretnénk. Az esetlegesen előforduló DC offszet miatt célszerű egy DC rezonátort is működtetni. Ebben az esetben a rezonátorok elhelyezkedése nem egyenletes, de ez a gyakorlatban nem okoz jelentős sebességcsökkenést. Többcsatornás esetben a mikrofonok számával megegyező független AC rezonátorpárra van szükség, és a szinusszal való gerjesztést meg kell ismételni mindegyik beavatkozó hangszóróból. A mérési eredményeket megfelelően rendezve megkapható az átviteli függvény-mátrix.

Többcsatornás zajelnyomás esetén az átviteli függvény-mátrix inverzét kiszámítva meghatározható a bemeneti analizátorkészletek és a kimeneti rezonátorkészletek közötti átcsatolást végző  $\mathbf{W}$  mátrix. Az  $\alpha$  konvergenciaparaméter állításával pedig a zajcsökkentő rendszer stabilitása biztosítható.

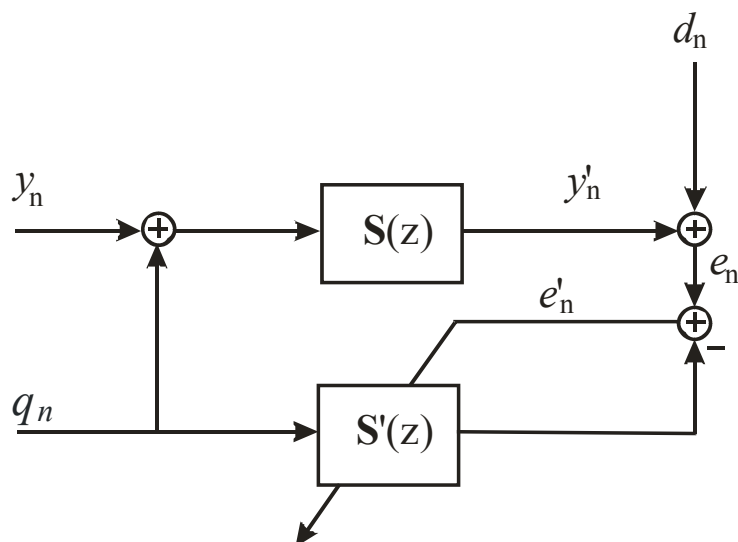
## 5. On-line identifikáció

Aktív zajcsökkentő rendszerek esetén a rendszer működése szempontjából kulcsfontosságú szerepet tölt be a másodlagos út (a beavatkozó hangszóróktól a hibamikrofonokig terjedő átvitel) átvitelének ismerete. A teljes zajcsökkentő rendszer akkor stabil, ha a másodlagos út modelljének a modellezési fázishibája nem haladja meg a  $\pi/2$  értéket. Sajnos az átviteli függvény időben változhat. A hőmérséklet vagy a páratartalom változása, egy szobában különböző tárgyak elmozdulása mind-mind befolyásolják a hibamikrofonok és a beavatkozó hangszórók közti akusztikus átvitelt. Ha ez a változás olyan mértékű, hogy az off-line meghatározott és a zajcsökkentésben alkalmazott átviteli függvény és inverz átviteli függvény már nem teljesíti a stabilitásra vonatkozó feltételt, akkor a rendszer instabillá válik. Ha valamilyen módon biztosítani tudnánk azt, hogy az off-line meghatározott modell megváltoztatásával a stabilitásra vonatkozó kritérium teljesüljön, akkor a rendszer továbbra is működőképes maradna. Ha tehát az átviteli függvényt folyamatosan, on-line identifikálni tudnánk, akkor ez a fenti probléma megoldását jelentené és egy stabil, robusztus zajcsökkentő rendszert kapnánk.

Ebben a fejezetben áttekintem a szakirodalomban megtalálható legfontosabb on-line identifikációs eljárásokat az előző fejezetekben bemutatott zajcsökkentő struktúrákra, bemutatva egy eddigiekben nem ismertetett, LMS algoritmus alapú zajcsökkentő rendszert is (Unified-error Filtered-X LMS algoritmus), valamint ismertetem az általam rezonátoros zajcsökkentő rendszerre kidolgozott on-line identifikációs eljárást.

### 5.1. On-line identifikáció Filtered-X, Filtered- $\epsilon$ és Filtered-EX LMS rendszerekben

Ebben a fejezetben bemutatom, hogy hogyan valósíthatjuk meg a másodlagos átviteli út identifikációját a Filtered-X, Filtered- $\epsilon$  és Filtered-EX LMS rendszerekben [1]. A három rendszer egy fejezetben tárgyalását az indokolja, hogy a szakirodalom mindhárom eljárásra ugyanazt az on-line identifikációs módszert ismerteti.



5.1. ábra. Dithert alkalmazó on-line identifikáció alapgongolata

Az on-line identifikáció alapgondolatát a Filtered-X, Filtered- $\epsilon$  és Filtered-EX LMS rendszerekben az 5.1. ábra szemlélteti. Az on-line eljárás mindhárom algoritmusba ugyanúgy épül be. Az ábrán  $y_n$  az eddigiekben  $W(z)$ -vel jelölt szűrő kimenetét,  $d_n$  az elsődleges út ( $P(z)$ ) kimenetét,  $e_n$  pedig a hibamikrofon jelét jelöli. Az elgondolás tehát az, hogy az ANC rendszerben a kontroller szűrő kimenetéhez folyamatosan, vagy csak bizonyos időközönként szélessávú zajt, úgynevezett dithert ( $q_n$ ) kell keverni, aminek a segítségével a hibamikrofon jelének felhasználásával a másodlagos út modelljét adaptálni lehet. Ez az algoritmus természetesen működőképes, előnye az, hogy jól követi a másodlagos út átvitelében bekövetkező változásokat. Hátránya azonban az, hogy a beavatkozó jelhez kevert dither nem nyomható el, így megnöveli a hibamikrofon környezetében az eredő zajteljesítményt. Erre természetesen adódik az a lépés, hogy csökkenteni kell a dither teljesítményét. Ezzel azonban az adaptáció sebességét rontjuk, és így a másodlagos út modellje nem lesz képes elég gyorsan követni az átvitelben beálló változásokat, ami akár a teljes rendszer instabilitását is okozhatja.

A Filtered- $\epsilon$  és Filtered-EX LMS rendszerekben a másodlagos átvitel identifikációján kívül szükség van még a késleltetett inverz előállítására is. Ez a Filtered-EX LMS algoritmus esetén könnyebb, mivel csak egy kis pontszámú (128-256) FFT-t és IFFT-t kell lefuttatni az on-line identifikáció mellett.

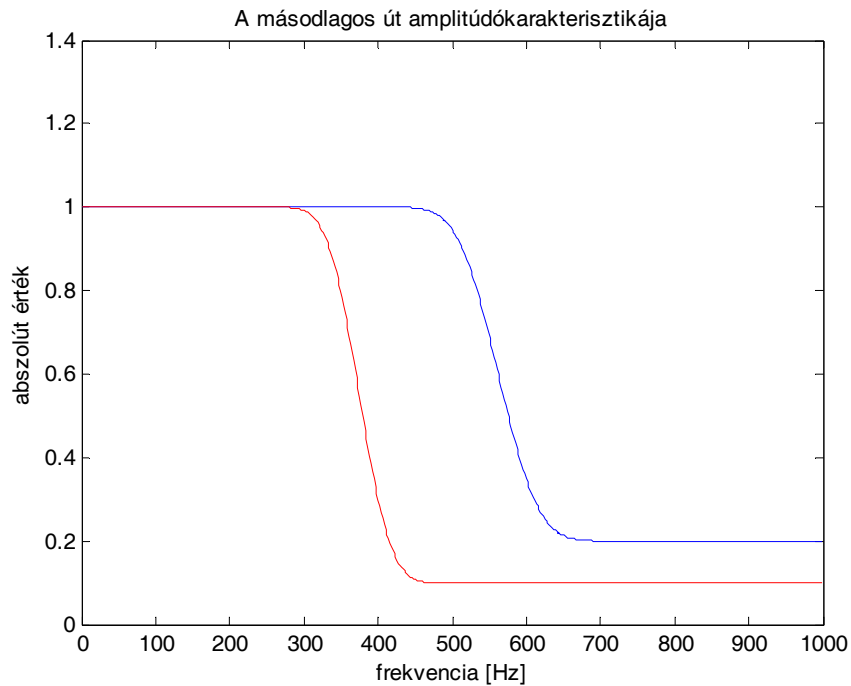
Az említett három rendszer közül kettő, a Filtered-X LMS és a Filtered-EX LMS algoritmus működését on-line identifikációval MATLAB környezetben is realizáltam. A Filtered- $\epsilon$  algoritmust azért hagytam ki a sorból, mivel konvergenciatulajdonságai körülbelül egyeznek a Filtered-EX algoritmuséval, tehát az on-line működésre is körülbelül hasonló eredmény várható, de a nehezkesebb számítási eljárás miatt megvalósításra nem választottam.

A szimulációban a könnyebb összehasonlítás végett ugyanazokat a beállításokat alkalmaztam. A mintavételi frekvencia 2 kHz volt, a szimulációs idő 40 s. A másodlagos utat modellező szűrő 200, a kontroller szűrő 405 együtthatós volt. A kontroller szűrő bátorsági tényezője  $\mu_1 = 0.0005$ , míg a másodlagos út on-line adaptációjának bátorsági tényezője  $\mu_2 = 0.005$  volt. Az elnyomandó jel normális eloszlású, zérus várható értékű, egységnyi varianciájú zaj volt. Dithernek ugyanilyen, csak kisebb teljesítményű zajt alkalmaztam.

Mivel ebben az esetben on-line identifikációt kellett szimulálni, a másodlagos út átvitelét időben változtattam. Az első 15 s-ban az átviteli függvény egy 62 együtthatós, Parks-McClellan algoritmussal tervezett lineáris fázismentű FIR szűrő volt. Az ezt követő 10 s alatt

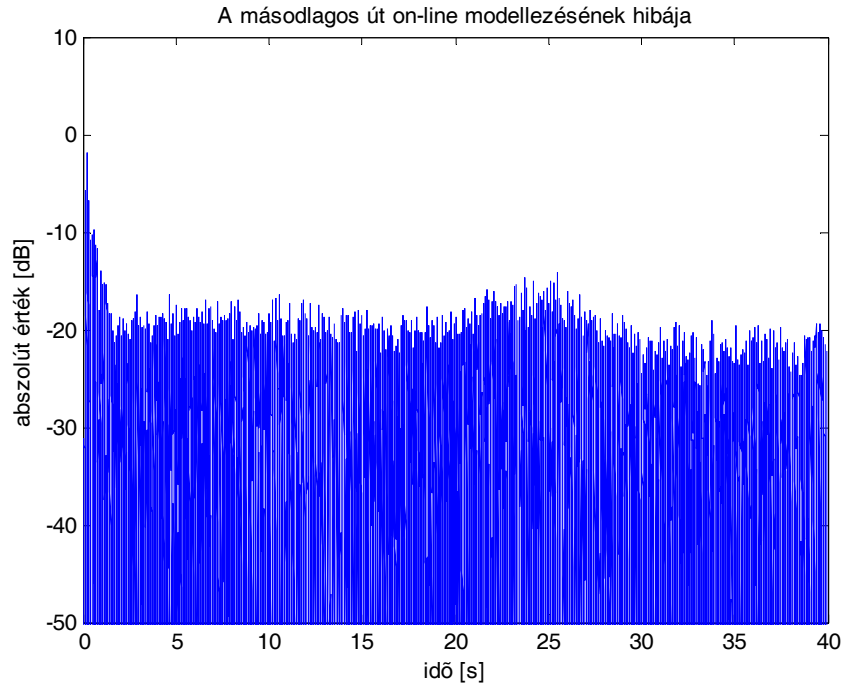


az átviteli függvény időben folyamatosan változott, a másodlagos út két szűrő lineáris kombinációja volt, és a súlyozás az idő haladtával lineárisan nőtt az egyik szűrő javára. Az egyik szűrő az előbbieken leírt szűrő volt, míg a másik egy ugyanilyen eljárással tervezett 70 együtthatós szűrő volt. A 10 s-os átmeneti időszak után az átviteli utat a második szűrő adta. Ezzel az eljárással sikerült modellezni a másodlagos út folyamatos változását. A fenti két szűrő amplitúdókarakterisztikája az 5.2. ábrán látható.

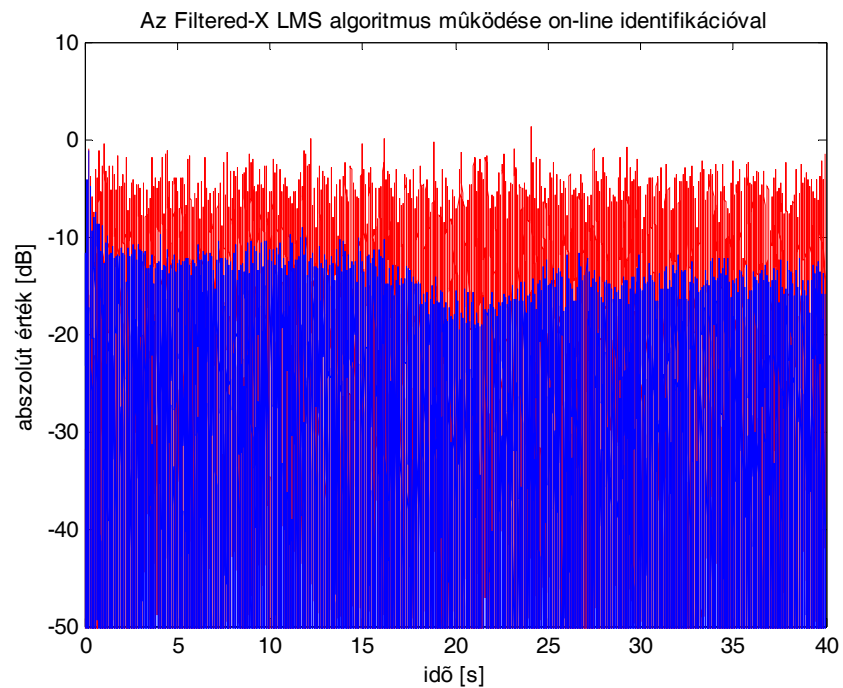


5.2. ábra. A másodlagos átviteli út amplitúdókarakterisztikája; mintavételi frekvencia 2 kHz

Kék szín jelöli azt az átvitelt, amiből kiindultam, míg piros, amelyikre változott. A következő két ábrán bemutatom a Filtered-X LMS algoritmus on-line változatának működését. Az 5.3. ábra a másodlagos út on-line modellezésének hibajelét, míg az 5.4. ábra a zajcsökkentést mutatja.



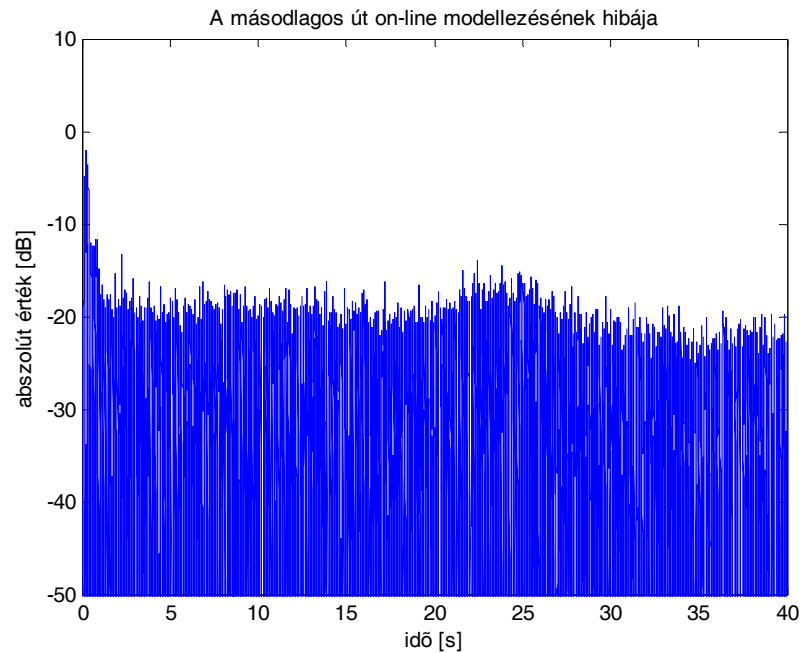
5.3. ábra. A másodlagos út on-line modellezésének hibája Filtered-X LMS algoritmusnál, az átviteli függvény a 15. s és a 25. s között változik; mintavételi frekvencia 2 kHz



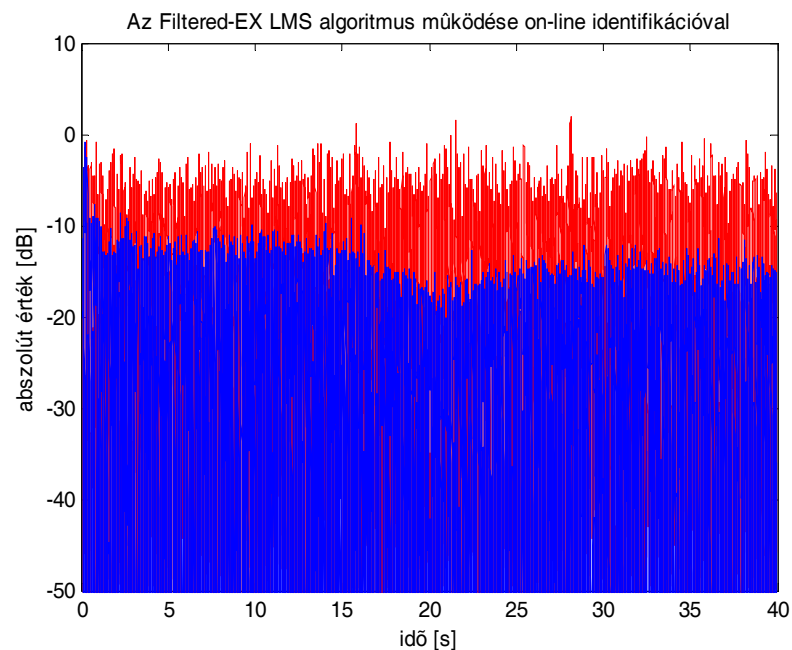
5.4. ábra. Filtered-X LMS algoritmus on-line identifikációval, az átviteli függvény a 15. s és a 25. s között változik; mintavételi frekvencia 2 kHz

Az 5.3. ábrán jól megfigyelhető, hogy a kezdeti állandósult állapotból a másodlagos út átviteli függvényének megváltozása megnöveli a modellezési hibát (ez a 15. s környékén észlelhető hibanövekedés), majd a modellezési hiba beáll a kezdeti állandósult állapotra jellemző szintre. A modellezési hiba azért nem csökkenhet az ábrán látható érték alá, mivel aktív zajcsökkentő alkalmazás révén a hibamikrofon jelét használjuk fel a másodlagos út on-line modellezésének számításához is, és ezért a hibamikrofon jele nem megszüntethető modellezési többletzajt okoz. Az 5.4. ábrán az elnyomás nélküli állapotot pirossal, míg a zajcsökkentés bekapcsolásával elért állapotot kékkel jelöltem. Az ábrán jól látszik, hogy a zajcsökkentő algoritmus működését az on-line identifikáció nem befolyásolja, kivéve azt a már korábban is említett hatást, hogy a dither maradó állandó zajteljesítményként jelenik meg a hibamikrofon jelében, ezért kapunk rosszabb elnyomási értéket, mint on-line identifikáció nélküli esetben. Természetesen az elnyomási tulajdonságokon lehetne javítani oly módon, hogy a dithert nem állandóan, hanem bizonyos időközönként adagolom a kimenethez. Erre két lehetőségünk van, de egyik sem ad optimális megoldást: az első, ha a dithert ritkán keverem a kimenethez, akkor az elnyomás nagyban javul, de a rendszer képtelen követni a gyors változásokat, amit akár egy ajtó kinyitása is okozhat, így a ritkán használt dither instabilitást eredményezhet. A második lehetőség, hogy nem állandóan adagolunk dithert, hanem viszonylag sűrűn; ezzel kiküszöbölhetjük az előző megoldás hátrányát, azaz képesek lehetünk követni a gyors változásokat, de mivel a dither hallható a hibamikrofonnál, ezért ez zavaró pulzáló hatást okoz, ami az ember számára zavaróbb, mint az állandóan hallható konstans teljesítményű zaj. Természetesen, ha nem kell felkészíteni a zajcsökkentő rendszert gyors változásokra, akkor a ritkán adagolt, nagy teljesítményű dither jó megoldást eredményezhet.

A következőkben a Filtered-EX LMS algoritmus szimulációját mutatom be ugyanazokkal a beállításokkal, amivel a Filtered-X LMS eljárás on-line működését szemléltettem. Az 5.5. és 5.6. ábrán tehát a Filtered-EX LMS algoritmus működését láthatjuk on-line másodlagos út modellezéssel.



5.5. ábra. A másodlagos út on-line modellezésének hibája Filtered-EX LMS algoritmusnál, az átviteli függvény a 15. s és a 25. s között változik; mintavételi frekvencia 2 kHz



5.6. ábra. Filtered-EX LMS algoritmus on-line identifikációval, az átviteli függvény a 15. s és a 25. s között változik; mintavételi frekvencia 2 kHz

Az 5.5. ábrán ugyanúgy, mint a Filtered-X LMS algoritmus esetében látható, hogy a másodlagos átviteli út változása a modellezési hibában növekedést eredményez. Az 5.6. ábra

nagyon hasonlít a Filtered-X LMS algoritmussal kapott eredményre. A két rendszer konvergencia-sebessége között most nem annyira szembeötlő a különbség, mint az on-line identifikáció nélküli esetekben. Ennek az oka az, hogy az elnyomás kisebb mértékű a dither miatt, valamint a kis teljesítményű dither miatt lecsökken a hurokerősítés, ezzel romlik mindkét rendszer konvergenciasebessége, így a két algoritmus sebessége közötti különbség nem annyira jelentős.

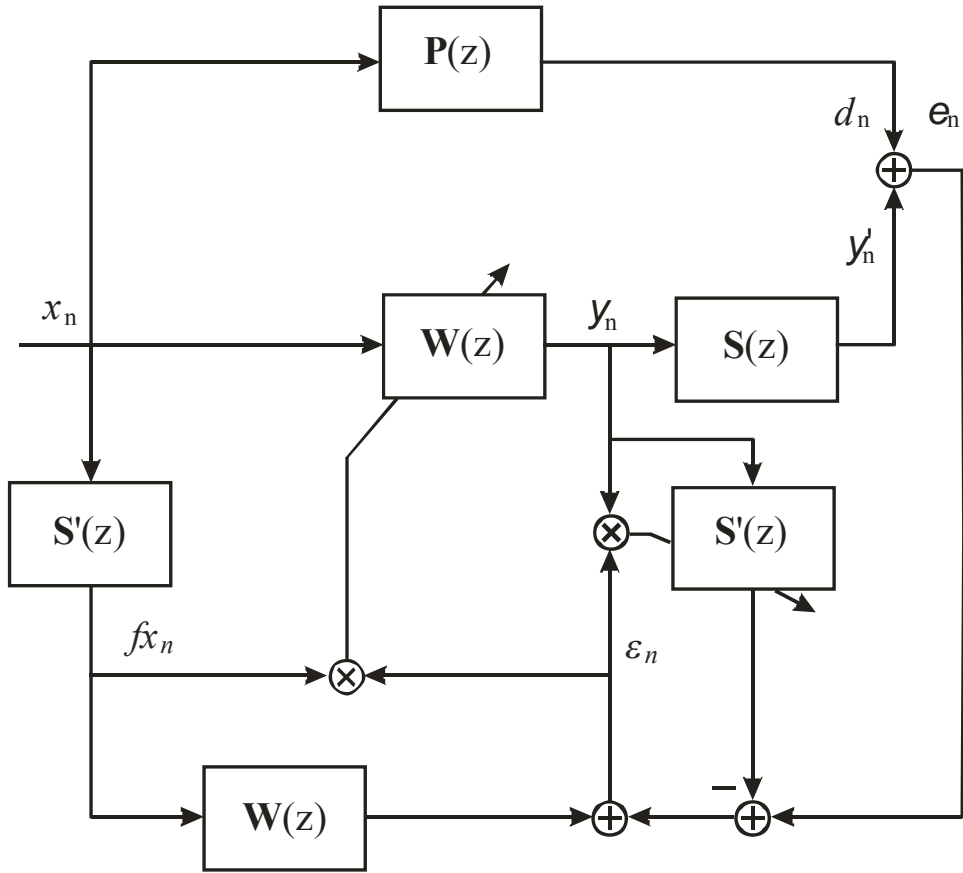
Láthattuk tehát, hogy a dithert alkalmazó megoldás robusztus rendszert kínál ugyan, de a dither hatása maradó hibaként a zajcsökkentés szempontjából nem előnyös, a dither tulajdonképpen éppen a célt, a zajcsökkentést akadályozza. Ezért a továbbiakban két olyan on-line algoritmust ismertetek, amelyek nem használnak dithert, mégis biztosítják a zajcsökkentés változó másodlagos átviteli út esetén is. Ezek közül az egyik az irodalomban fellelhető, míg a másik saját ötleten alapuló algoritmus.

## ***5.2. On-line identifikáció Unified-error Filtered-X LMS algoritmussal [4]***

Ahogy az előző fejezetben láttuk, dither adagolása a beavatkozó jelhez on-line identifikáció céljából teljesítményromláshoz vezet a zajcsökkentés szempontjából. Ráadásul, hogy a zajcsökkentést ne rontsuk el, kis teljesítményű dithert használunk, ami azt eredményezi, hogy az on-line adaptáció nagyon lassú lesz, így a gyors változásokat esetleg nem tudja jól követni, ami rossz esetben az egész rendszer instabilitását eredményezheti.

A fenti problémát oldja meg a Unified-error Filtered-X LMS algoritmus. Bevezetésképpen (a pontos tárgyalás később következik): algoritmus onnan kapta a nevét, hogy egyesíti a modellezési és a szabályozási hibát egy, úgynevezett egyesített, egységes (angolul: unified) hibába. Ezáltal a modellezési és a szabályozási folyamatot nem választja szét, hanem egyesíti annak érdekében, hogy egy nagyobb konvergenciasebességű, a gyors változásokat követni képes rendszert kapjunk.

A Unified-error Filtered-X LMS algoritmus blokkvázlata az 5.7. ábrán látható.



5.7. ábra. A Unified-error Filtered-X LMS algoritmus blokkvázlata

Az ábráról jól látszik, hogy a Unified-error Filtered-X LMS algoritmus ugyanúgy két adaptív szűrővel dolgozik, mint az előző fejezetben bemutatott ditherrel dolgozó on-line algoritmusok. A legnagyobb különbség a kétfajta megközelítés között ott van, hogy míg a ditherrel működő on-line algoritmusok teljesen külön kezelik a másodlagos út identifikációját és a szabályozást (zajcsökkentést), addig a unified-error algoritmus a két feladatot párhuzamosa, összevonva kezeli.

Az algoritmus működése képletekkel is megfogalmazható. Ez a következőképp érhető el:

$$f\hat{x}_k = \sum_{j=1}^{L_{S'}} s'_{k_j} x_{k-j} \quad (5.1)$$

$$\varepsilon_k = e_k - \sum_{j=1}^{L_{S'}} s'_{k_j} y_{k-j} + \sum_{i=0}^{L_W} w_{k_i} f\hat{x}_{k-i} \quad (5.2)$$

$$w_{k+1_i} = w_{k_i} - 2\mu\varepsilon_k f\hat{x}_{k-i} \quad (5.3)$$

$$s'_{k+1_j} = s'_{k_j} + 2\mu\varepsilon_k y_{k-j} \quad (5.4)$$

$$y_k = \sum_{i=0}^{L_W} w_{k_i} x_{k-i} \quad (5.5)$$

Mind a  $w$ , mind az  $s'$  együtthatókat 0-ra kell inicializálni, az  $\mathbf{fx}$  együtthatóvektort pedig fel kell tölteni kis teljesítményű zajmintákkal.

Azt azonban meg kell jegyezni, hogy a unified-error algoritmus szempontjából nem számít, hogy a másodlagos út identifikációja sikeres-e vagy nem, mivel a másodlagos út modellezése az algoritmusnak csupán másodlagos problémája, az eljárás elsődleges célja a nagy elnyomás és a nagy konvergenciasebesség biztosítása.

### 5.2.1. Analízis

A következőkben bemutatásra kerül, hogyan egyesíti a unified-error algoritmus a modellezési és a szabályozási hibát, majd elvégzem a unified-error algoritmus stabilitásanalízisét és bemutatom az eljárás normalizálását. Végül a fejezetet MATLAB környezetben végzett szimulációs eredményeim bemutatásával zárom.

Az egyesített hiba magyarázatához tegyük fel, hogy a  $w^*$  együtthatók az ideális kontroll szűrő együtthatói, azaz kielégítik az (5.6) egyenletet:

$$[S(z)W^*(z) + P(z)]X(z) = 0 \quad (5.6)$$

Valamint legyenek az  $s^*$  együtthatók az ideális  $s'$  szűrő együtthatói, azaz kielégítik az (5.7) egyenletet:

$$[S^*(z)W(z) - S(z)\{W(z) - W^*(z)\} - W^*(z)S'(z)]X(z) = 0 \quad (5.7)$$

Most nézzük az egyesített hiba kifejezését:

$$Z\{\varepsilon_k\} = [P(z) + S(z)W(z) - S'(z)W(z) + W(z)S'(z)]X(z) \quad (5.8)$$

Amiből az (5.6) és az (5.7) egyenletek felhasználásával:

$$Z\{\varepsilon_k\} = [W(z) - W^*(z)]S'(z)X(z) - [S'(z) - S^*(z)]W(z)X(z) \quad (5.9)$$

Az (5.9) egyenletből pedig kiolvasható, hogy az egyesített hiba valóban tartalmazza mind a modellezési, mind a kontroll hibát.

A következőkben a unified-error algoritmus Lyapunov-féle stabilitásanalízisét mutatom be. Először definiálni kell egy pozitív semi-definit Lyapunov függvényt, ami esetünkben az (5.10) egyenlettel adott [4]:

$$V_k = \frac{1}{\mu_w} \sum_{i=0}^{L_w} (w_{k_i} - w_i^*)^2 + \frac{1}{\mu_{s'}} \sum_{i=0}^{L_w} (s'_{k_i} - s_i^*)^2 \quad (5.10)$$

Az (5.3) és (5.4) egyenleteket felhasználva megalkothatóak az (5.11) és (5.12) egyenletek:

$$(w_i - w_i^*)_{k+1} = (w_i - w_i^*)_k - 2\mu\varepsilon_k f x_{k-i} \quad (5.11)$$

$$(s'_i - s_i^*)_{k+1} = (s'_i - s_i^*)_k + 2\mu\varepsilon_k y_{k-j} \quad (5.12)$$

Az (5.11) és (5.13) egyenleteket felhasználva pedig egy rekurzív formula adódik  $V_k$ -ra:

$$V_{k+1} = V_k + 4\varepsilon_k^2 \left\{ \left[ \mu_w \sum_{i=0}^{L_w} f x_{k-i}^2 + \mu_{s'} \sum_{i=1}^{L_{s'}} y_{k-i}^2 \right] - 1 \right\} \quad (5.13)$$

Ha a  $\mu_w$  és  $\mu_{s'}$  helyett csak egyetlen bátorsági tényezőt alkalmazunk ( $\mu$ ), akkor az (5.13) egyenlet akkor lesz konvergens, ha (5.14) teljesül:

$$0 < \mu < \frac{1}{\sum_{i=0}^{L_w} f x_{k-i}^2 + \sum_{i=1}^{L_{s'}} y_{k-i}^2} \quad (5.14)$$

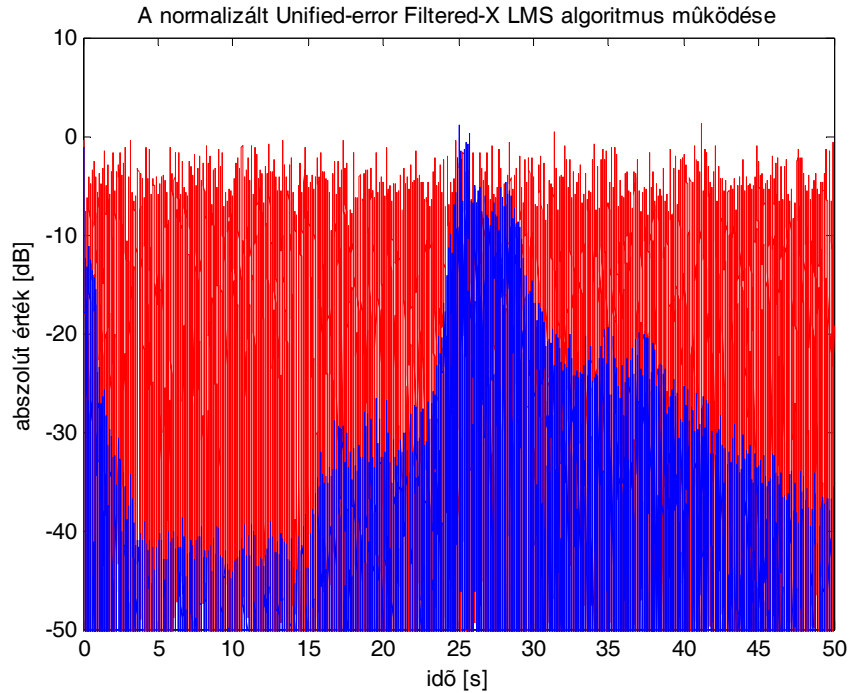
Az algoritmus normalizálása azt jelenti, hogy az alkalmazott bátorsági tényezőt minden esetben leosztjuk az (5.14) egyenlőtlenség jobb oldalán a nevezőben található be- és kimeneti jelek teljesítményére jellemző számmal, azaz  $N_k$ -val:

$$N_k = \sum_{i=0}^{L_w} f x_{k-i}^2 + \sum_{i=1}^{L_{s'}} y_{k-i}^2 \quad (5.15)$$

Ha ezt az eljárást követjük, akkor a bátorsági tényezőnk minden mintavételi időpontban a bemenő jel teljesítményének megfelelően alakul, így az algoritmus stabilitása és a megfelelő konvergenciasebesség biztosítható.

A fentiekben bemutatott normalizált Unified-error Filtered-X LMS algoritmus működését MATLAB környezetben szimuláltam. A szimulációs beállítások ugyanazok voltak, mint az előző fejezetben vizsgált ditherrel dolgozó on-line algoritmusoknál. Az algoritmus inicializálásánál a  $\mathbf{w}$  vektor együtthatóit nulláról, az  $\mathbf{s}'$  vektor együtthatóit pedig egy off-line identifikált állapotról indítottam. Az  $\mathbf{f}\mathbf{x}$  vektort pedig kis teljesítményű fehér zaj mintákkal töltöttem fel. A szimulációs eredmény a zajcsökkentésre az 5.8. ábrán látható. Az ábrán pirossal a zajcsökkentés nélküli, kékkkel a zajcsökkentéssel elért állapotot jelöltem.





5.8. ábra. A Unified-error Filtered-X LMS algoritmus működése, az átviteli függvény a 15. s és a 25. s között változik; mintavételi frekvencia 2 kHz

Az ábráról jól látszik, hogy a másodlagos átviteli út megváltozása, ami a 15. s-tól kezdődik, az elnyomásban is érezteti hatását. A hibajel megnő, de a változás után nagy meredekséggel csökken egy állandósult állapotbeli érték felé. A szimuláció első másodperceiben a hibajel igen gyorsan lecsökken, aminek a magyarázata az, hogy az  $s'$  vektor együtthatóit nem nulláról, hanem egy előre identifikált állapotról indítottam azért, hogy a szimulációban jól láthassuk az állandósult állapot, és az átviteli függvény változásakor bekövetkező elnyomás változása közti különbséget, valamint így könnyebb összehasonlítani az algoritmust a ditherrel dolgozó on-line identifikációval működő Filtered-X LMS és Filtered-EX LMS algoritmusokkal. Az eredmény alapján jól látszik, hogy a unified-error algoritmus állandósult állapotban körülbelül akkora elnyomást tud, és olyan sebességgel, mint az Filtered-EX LMS. A másodlagos átviteli út változásakor ez az elnyomás romlik, de a változás után néhány másodperccel már sokkal nagyobb elnyomást biztosít, mint a dithert használó megoldások. Ezért választottam megvalósítás céljára az on-line identifikációval működő LMS alapú zajcsökkentő rendszerek közül a Unified-error Filtered-X LMS algoritmust.

### 5.3. *On-line identifikáció rezonátoros rendszerekben*

A rezonátoros struktúrával való aktív zajcsökkentés esetén nagy szerepe van az átviteli függvény mátrix off-line identifikálásának, hiszen ennek alapján határozható meg az inverz átviteli függvény mátrix, aminek alkalmazásával a rezonátoros struktúra visszacsatoló ágában  $A(z)=1$  értékű visszacsatolás biztosítható a rezonátorfrekvenciákon. Sajnos az átviteli függvény időben változhat. A hőmérséklet vagy a páratartalom változása, egy szobában különböző tárgyak elmozdulása mind-mind befolyásolják a hibamikrofonok és a beavatkozó

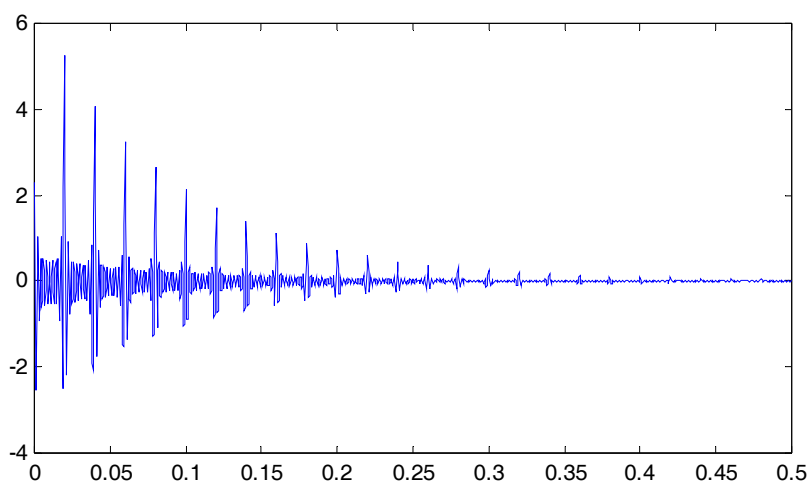
hangszórók közti akusztikus átvitelt. Ha ez a változás olyan mértékű, hogy az off-line meghatározott és a zajcsökkentésben alkalmazott inverz átviteli függvény mátrix ( $\mathbf{W}$ ) már nem teljesíti a stabilitásra vonatkozó, (4.28) egyenletben megfogalmazott feltételt, akkor a rendszer instabillá válik. Ha valamilyen módon biztosítani tudnám azt, hogy az együtthatók megváltoztatásával a (4.28) egyenlet teljesüljön, akkor a rendszer továbbra is stabil maradna. Ha tehát az átviteli függvényt folyamatosan, on-line identifikálni tudnám, akkor ez a fenti probléma megoldását jelentené és egy stabil, robusztus zajcsökkentő rendszert kapnánk. Ez tehát egy még nem megoldott probléma, ebben a fejezetben ismertetem az általam egycsatornás rendszerre kidolgozott megoldást. Az előzőekben már bemutattam, hogy milyen hátrányos következményei vannak a dither alkalmazásának. A gerjesztés alkalmazásának elkerülésére dolgoztam ki egycsatornás rendszerre az általam on-line identifikációs algoritmusnak nevezett eljárást. Ez a módszer nem alkalmaz gerjesztést, de tulajdonképpen nem is identifikációs eljárás. Ezzel a módszerrel a rendszer stabilitása és ezáltal a zajelnyomás biztosítható.

### 5.3.1 On-line identifikáció egycsatornás rendszerre

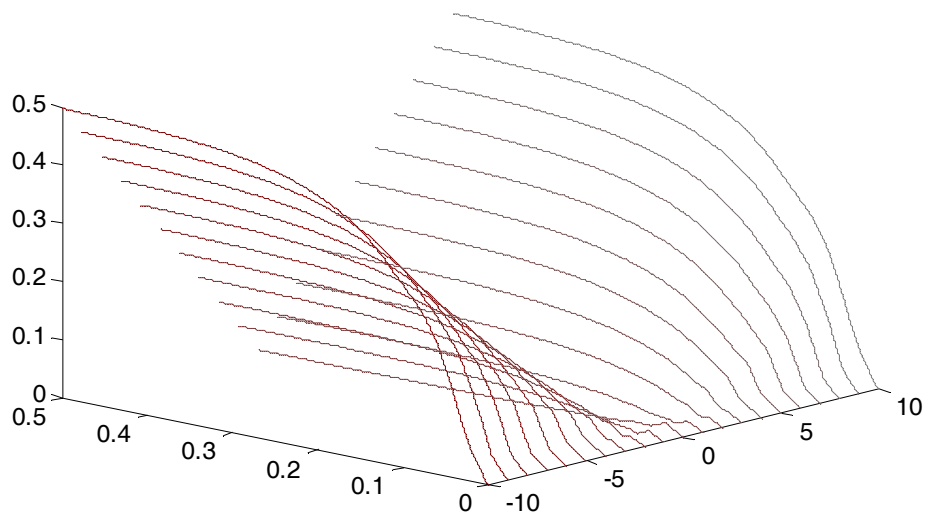
A tanszéken felvetődött ötlet az volt, hogy ha egy egycsatornás rendszer instabil, akkor az azt jelenti, hogy a rendszer valamely  $w_i$  becsatoló együtthatói nem teljesítik az (4.23) egyenletben megfogalmazott feltételt. Ekkor úgy tehetjük stabillá a rendszert, hogy az instabilitást okozó  $w_i$  együtthatók fázisát  $180^\circ$ -kal elforgatjuk.

Ahhoz, hogy ezt a beavatkozást elvégezhessük, meg kell tudnunk állapítani, hogy a zajcsökkentő rendszer instabil-e, és ha igen, akkor mely becsatoló együtthatók okozzák ezt az állapotot. A beavatkozási módszer tehát adott volt, ezután egy megfelelő algoritmust kellett találnom arra, hogy miként állapítsuk meg azt, hogy mely becsatoló együtthatók okozzák a rendszer instabilitását.

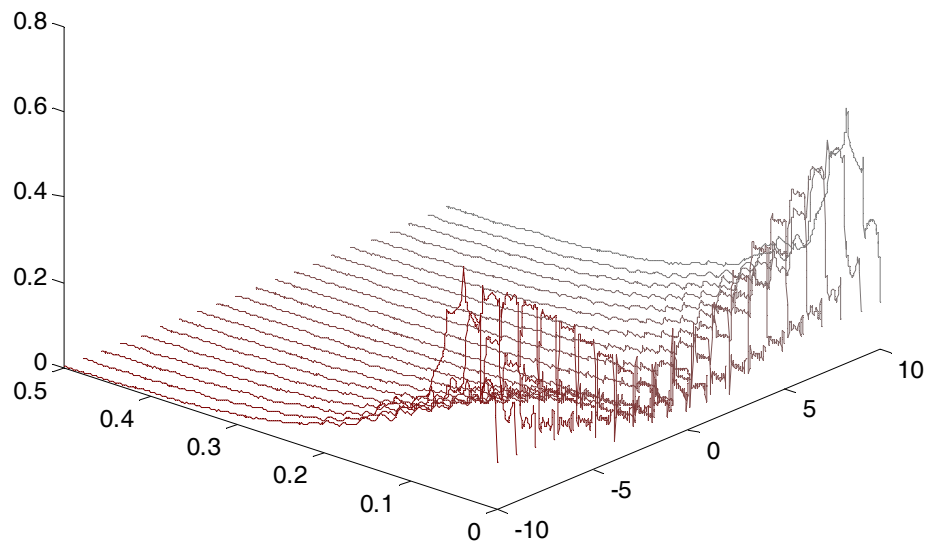
A probléma megoldására MATLAB-ban szimuláltuk egy egycsatornás zajcsökkentő rendszer viselkedését. A következő ábra egy stabil rendszer működését mutatja:



5.9. ábra. Hibajel alakulása az idő függvényében, stabil rendszer esetén; mintavételi frekvencia 2 kHz



5.10. ábra. Kimeneti rezonátorkészlet tárolók abszolút értéke az idő függvényében, stabil rendszer esetén; mintavételi frekvencia 2 kHz



5.11. ábra. A hibajel Fourier-komponensekre bontó rezonátorkészlet rezonátorai abszolút értékének alakulása az idő függvényében, stabil rendszer esetén; mintavételi frekvencia 2 kHz

A rendszerben a mintavételi frekvencia 2 kHz. A szimuláció során nem alkalmaztam adaptív Fourier-analízist, az elnyomandó jelet úgy generáltuk, hogy annak komponensei a rezonátor-frekvenciákra essenek. A rendszerben 21 rezonátor van, ebből 10 pár AC, és 1 DC rezonátor. Ez igaz mind az analízátorkészletre, mind a kimeneti rezonátorkészletre. A rezonátorok 50 Hz-enként helyezkednek el. Az akusztikus átvitelt egy másodfokú IIR szűrővel helyettesítettem, melynek átviteli függvénye igen jól szelektál az algoritmusok között. Átviteli függvénye a következő:

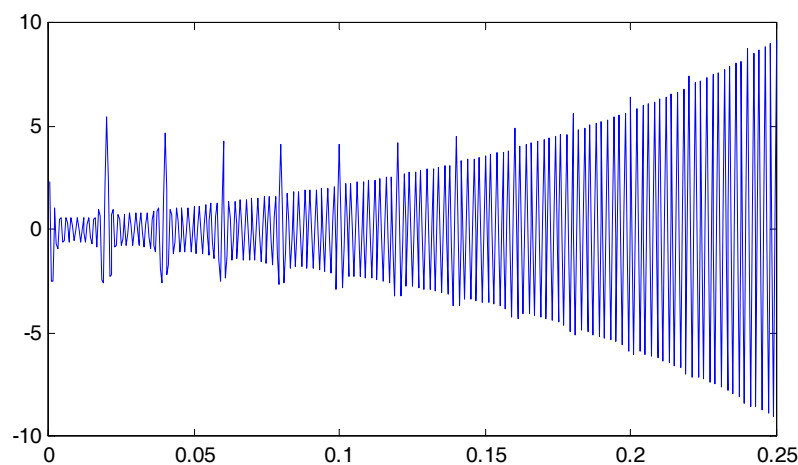
$$W(z) = \frac{z^2 - 0.4164z + 1.2346}{z^2 + 0.6627z + 0.6414} \quad (5.16)$$

Az elnyomandó jel a következő képlettel adott:

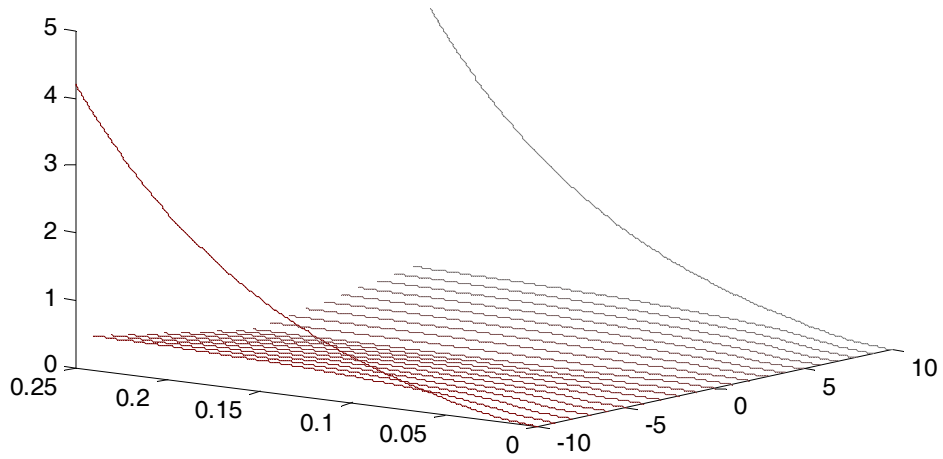
$$n_n = \sum_{i=1}^{10} \cos(2\pi i \frac{50}{2000} n) \frac{i}{10} \quad (5.17)$$

A 5.9-es ábra a hibamikrofon jelének alakulását mutatja az idő függvényében. Az ábrán jól látható, ahogy a hibajel exponenciálisan csökken. A 5.10-es ábra a kimeneti rezonátorkészlet rezonátorai abszolút értékének alakulását mutatja az idő függvényében, az összes rezonátorra. Látszik, hogy a rezonátorok rövid idő alatt beállnak egy konstans értékre, amit a továbbiakban tartanak. A 5.11-es ábra a hibajelet Fourier-komponensekre bontó rezonátorkészlet rezonátorai abszolút értékének alakulását mutatja az idő függvényében, az összes rezonátorra. Az ábráról leolvasható, hogy az analízátorkészlet összes rezonátora rövid idő alatt beáll a 0 értékre, azaz a hiba minimalizálódik.

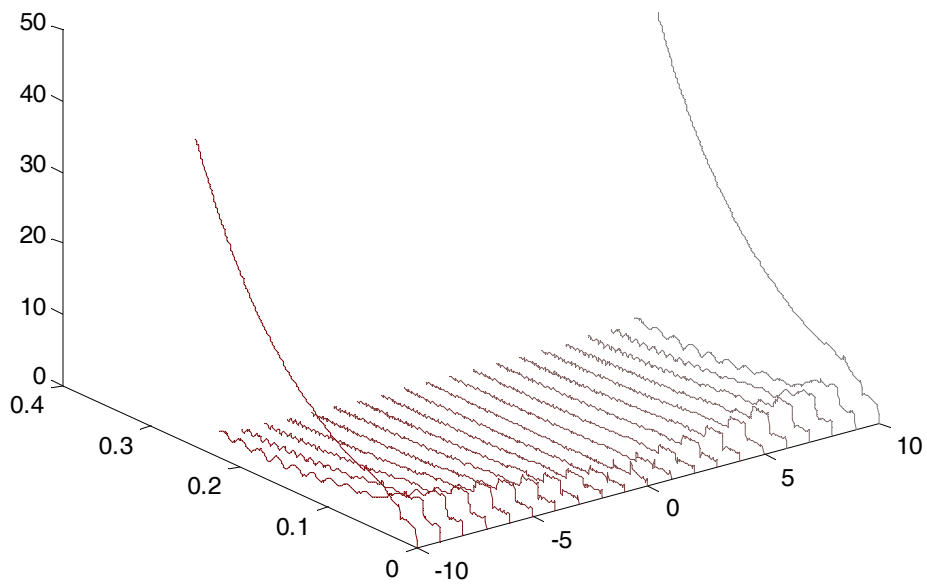
A következő ábrákon ugyanazokat ábrázoltuk, mint az előbbieken, azzal a különbséggel, hogy az  $f=500$  Hz-en működő rezonátor pár becsatoló együtthatóit elforgattam  $180^\circ$ -al, ez ekvivalens azzal, mintha csak  $f=500$  Hz-en változott volna meg az akusztikus átvitel, méghozzá úgy, hogy az átviteli függvény fázisa  $f=500$  Hz-en  $180^\circ$ -ot fordult. A szimulációs eredmények a következők:



5.12. ábra. Hibajel alakulása az idő függvényében, instabil rendszer esetén; mintavételi frekvencia 2 kHz



5.13. ábra. Kimeneti rezonátorkészlet tárolók abszolút értéke az idő függvényében, instabil rendszer esetén; mintavételi frekvencia 2 kHz

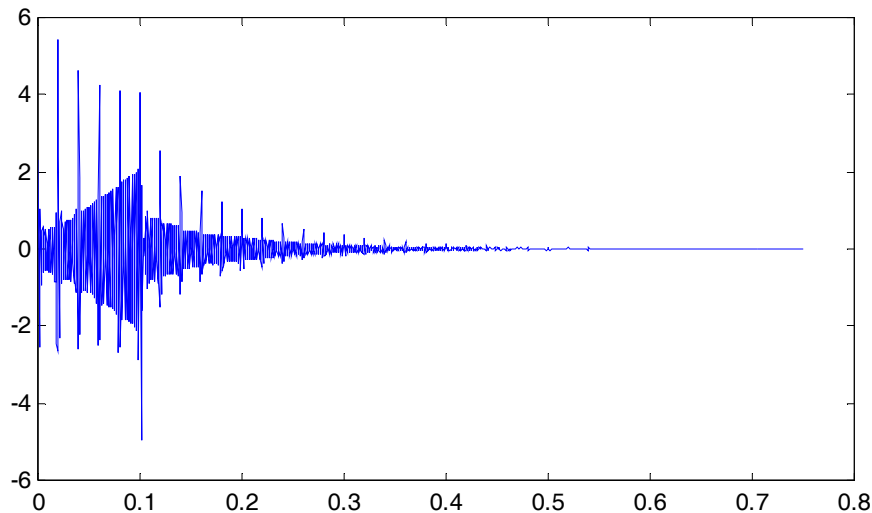


5.14. ábra. A hibajelel Fourier-komponensekre bontó rezonátorkészlet rezonátorai abszolút értékének alakulása az idő függvényében, instabil rendszer esetén; mintavételi frekvencia 2 kHz

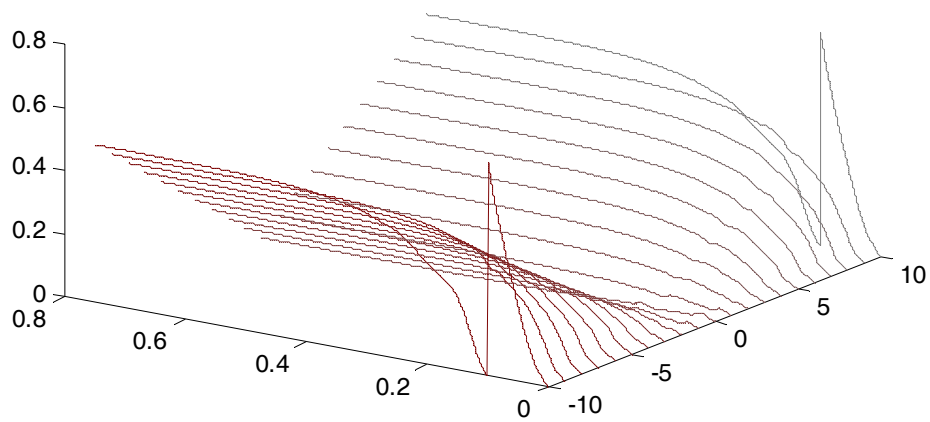
Az ábrákon jól látható, hogy a legnagyobb frekvenciás rezonátor pár tárolóinak abszolút értéke mind a bemeneti rezonátorkészletben, mind a kimeneti rezonátorkészletben exponenciális növekedésnek indul. Ezzel együtt a hibajel is exponenciálisan növekszik.

A stabil és az instabil állapot közötti különbséget vizsgálva azt vettük észre, hogy ha a rendszer stabil, akkor a kimeneti rezonátorkészlet tárolóinak abszolút értéke kezdetben nagyon gyorsan változik, majd később időegység alatt egyre kevesebbet. Ez azt jelenti, hogy stabil rendszer esetén a kimeneti rezonátorkészlet tárolói abszolút értékének deriváltja az idő függvényében csökken. Ellenben, ha a rendszer instabil, akkor a kimeneti rezonátorkészlet tárolóinak abszolút értéke exponenciálisan növekszik, azaz időegység alatt egyre többet változik, tehát deriváltja növekvő. Ezt a jelenséget használtam fel a rendszer instabilitásának detektálására. Egycsatornás esetben a bemeneti analizátorkészlet tárolóinak abszolút értéke a kimeneti rezonátorkészlet tárolói abszolút értékének a deriváltjával egyenlő.

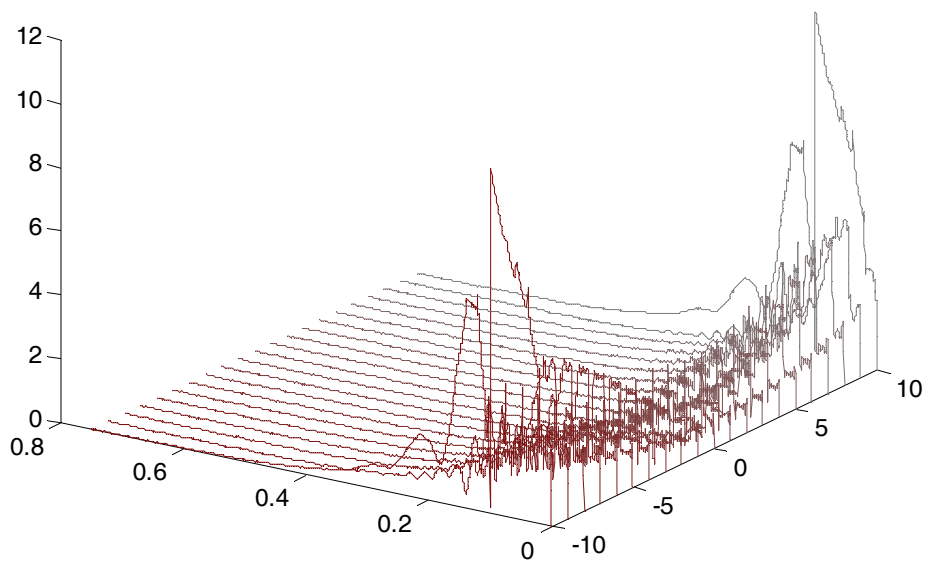
A hibadetektáló algoritmus tehát az, hogy figyelem a kimeneti rezonátorkészlet tárolóinak deriváltját, és ha ez növekvő, akkor beavatkozunk és az adott  $w_i$  becsatóló együttható fázisát elforgatunk  $180^\circ$ -kal. Előfordulhat azonban, hogy a stabilitás határhelyzetében levő rendszerből a megfelelő  $w_i$  együtthatón végrehajtott  $180^\circ$ -os fázisforgatás szintén a stabilitás határhelyzetében lévő rendszert hoz létre. Ennek elkerülésére, ha az algoritmus azt észleli, hogy a rendszer nem stabil, akkor először  $180^\circ$ -kal elfogatja a megfelelő  $w_i$  együttható fázisát. Ha az így kapott rendszer instabil, akkor ugyanakkor a  $w_i$  együtthatónak a fázisát  $20^\circ$ -kal forgatja el. A  $180^\circ$ - és  $20^\circ$ -os fázisforgatást ugyanarra a  $w_i$  együtthatóra felváltva alkalmazza az algoritmus. A tapasztalat azt mutatta, hogy a stabil állapotot a rendszer a fent említett két forgatás változtatásával éri el a leggyorsabban. Az algoritmus működését a következő három ábrán mutatjuk be. A helyzet ugyanaz, mint az előző esetben, azaz az  $f = 500$  Hz-en működő rezonátorpár becsatóló együtthatói rosszak.



5.15. ábra. Hibajel alakulása az idő függvényében, instabil, de javított rendszer esetén; mintavételi frekvencia 2 kHz



5.16. ábra. Kimeneti rezonátorkészlet tárolók abszolút értéke az idő függvényében, instabil, de javított rendszer esetén; mintavételi frekvencia 2 kHz



5.17. ábra. A hibajel Fourier-komponensekre bontó rezonátorkészlet rezonátorai abszolút értékének alakulása az idő függvényében, instabil, de javított rendszer esetén; mintavételi frekvencia 2 kHz

Az ábrákon jól látható, hogy a hiba egy ideig nő, ugyanígy nő exponenciálisan az  $f = 500$  Hz-en működő rezonátorpár tárolóinak abszolútértéke mind a kimeneti rezonátorkészletben, mind a bemeneti analizátorkészletben. Majd a hibadetektáló algoritmus

észreveszi, hogy a rendszer instabil és beavatkozik. Az eredmény: a beavatkozást követően a tárolók beállnak egy konstans értékre, a hibajel lecsökken, a rendszer stabilná vált.



## 6. A feladatmegoldás hardware és software eszközei

### 6.1. Az ADSP-21061 EZ-KIT Lite fejlesztőkártya

Az ADSP-21061 EZ-KIT Lite az ADSP-21061 jelfeldolgozó processzor köré van építve. Ezek a kártyák mentesítik a vásárlót a tesztáramkör építése alól, ami például egy lebegőpontos DSP esetében négyrétegű nyomtatott áramkör tervezését jelenti. Ehelyett a vásárló egy olyan kártyát kap kézbe, amelynek segítségével az adott termék rögtön hozzáférhető és működés közben kipróbálható. Ez nagymértékben elősegíti az adott alkatrész gyors megismerését.

Az ADSP-21061 EZ-KIT Lite fejlesztői kártya elsődlegesen arra szolgál, hogy az ADSP-21061 processzor minél több funkciója elérhetővé váljon, egyszerűen hozzáférhető legyen. A kártyán négy LED és három nyomógomb található. A LED-eket a DSP flag lábai hajtják meg. A három nyomógomb közül egy a Reset lábra, egy a flag1 lábra, egy pedig az IRQ1 lábra van illesztve. Ez utóbbi megnyomásakor megszakításkérés történik, amely lehetőség nagyon jól kihasználható a különböző alkalmazások implementálásakor.

A kártyán található portok egyike a JTAG emulátor port, amelyen keresztül hibakeresésre (debugging) nyílik lehetőség. A JTAG port használatáról bővebben a későbbiekben térek ki.

A DSP külső memóriabuszára a logikai áramkörön kívül egy gyárilag beégetett EPROM-ot is illesztettek. A processzor ennek segítségével hajtja végre a bootolási szekvenciát hardver reset után.

Egy AD1847-es sztereó codec-et is elhelyeztek a kártyán, amelyhez analóg oldalról két sztereó jack dugóval lehet csatlakozni (egy kimenet, egy bemenet). Ez két bementi és két kimeneti csatornát jelent. Az AD1847-es codec a DSP nulladik soros portjához (SPORT0) van illesztve.

A fejlesztői kártyán a DSP azon lábaihoz is vannak kivezetések, amelyekhez nincs alkatrész illesztve. Ezek a lábak akkor válnak elérhetővé, ha tűskesorokat forrasztunk a megfelelő helyekre.

### 6.2. ADSP -21061

A fejlesztői kártya legfontosabb eleme a DSP, a jelfeldolgozó processzor. Az ADSP-21061 a SHARC DSP-család egyik tagja, amely az ADI egyik leginkább elterjedt és széles körben használt termékcsaládja. A SHARC betűszó a Super Harvard ARchitecture Computer rövidítése. Ez a felépítés a DSP-k körében klasszikusnak számító Harvard-architektúra továbbfejlesztett változata. A Harvard-architektúra a személyi számítógépek processzorainak Neumann-architektúrájával szemben megkülönböztet kód- illetve adatmemóriát. A két memóriaterület külön buszon érhető el, így lehetőség van párhuzamos használatukra. Ez a lehetőség műveletek ciklikus elvégzésekor nagymértékben gyorsítja a program futását.

A szuper Harvard-architektúra esetében már négy független busz van: két adatbusz, egy utasításbusz és egy IO busz. A SHARC processzorok egy órajelütemben egy utasítást hajtanak végre, ezt a token belüli megvalósítású utasítás-cache teszi lehetővé.

Az ADSP-21061 a SHARC család első tagjának, az ADSP-21000-nek egy továbbfejlesztett változata. A processzor mag (processor core) megegyezik a két típusnál, de a

21061-es néhány fontos kiegészítővel látták el. Ezek a következők: SRAM, IO processzor, IO busz. Ezen egységek mindegyike tokon belül van megvalósítva.

A 21061-es nem a legfejlettebb modell a ma kapható DSP-k között, de sebessége és számítási pontossága kielégítő a legtöbb átlagos jelfeldolgozási alkalmazás szempontjából. Az iparban széles körben használják ezt a DSP-t, mert viszonylag alacsony ára mellett az összes olyan követelményt teljesíti, ami egy lebegőpontos jelfeldolgozó processzortól elvárható.

A távolabbi jövőben várható, hogy a SHARC család tagjait fel fogják váltani az újabb DSP családok, például a ma már kapható, de még ritkaságszámba menő TigerSHARC család tagjai.

### 6.2.1. Teljesítményadatok

A CMOS megvalósítású ADSP-21061 processzor maximális órajelfrekvenciája 50 MHz. Mivel a DSP órajelütemenként egy utasítást hajt végre az utasítás cache segítségével, így az utasításvégrehajtási sebesség 50 MIPS (Million Instructions Per Second).

### 6.2.2 Felépítés

A processzor fő részei a következők:

#### ▪ **Műveletvégző egység**

A műveletvégző egység kétféle lebegőpontos számábrázolási formátum használatát támogatja, a 32 bites IEEE single-precision floating-point szabványos formátumot, illetve a 40 bites kiterjesztett (extended precision) formátumot. Lehetőség van 32 bites fixpontos formátum használatára is. A műveletvégző egység részei: ALU (Arithmetic and Logic Unit), szorzó egység, shifter egység. Ezek párhuzamosan vannak megvalósítva, tehát lehetőség van egyidejű használatukra a műveletvégzés gyorsítása érdekében.

Az egység számítási kapacitása általános esetben a fentebb említett 50 MIPS. Ez azonban egyes tipikus jelfeldolgozási számítások során a műveletek párhuzamos végrehajtása miatt megnövekszik. Az ilyen módon elérhető maximális számítási kapacitás 120 MIPS vagy 120 MFLOPS (Million Floatingpoint Operation Per Second). Ez utóbbi mérőszám is gyakran használatos, a processzor lebegőpontos számítási műveletekre vonatkoztatott számítási sebességet jellemzi. Mivel az ADSP-21061-es alapértelmezésben lebegőpontos számokon hajtja végre az utasításokat, esetében a két mérőszám megegyezik.

#### ▪ **Adatregiszter blokk**

32 általános célú regiszter áll rendelkezésre a különböző adatmozgatási és számítási feladatokhoz. Ez a 32 regiszter két 16-os csoportra oszlik, az elsődleges (primary) és a másodlagos (alternate) regiszterblokkokra. A két blokk közül mindig csak az egyik aktív, a másik nem hozzáférhető. A blokkok közti váltás egy utasítással történik. Ezt a funkciót tipikusan a megszakítási rutinok használják ki, olyan módon, hogy az IT rutin csak a másodlagos regiszterblokkot használja, a főprogram pedig az elsődlegest. Ekkor elkerüljük azt, hogy a két rutinnak közös regisztereken kelljen osztoznia. Tekintve, hogy a megszakítási rutinok a főprogramot tetszőleges helyen szakíthatják meg, a közös regiszterhasználat előre nem tervezhető konfliktusokat okozna. Ez a probléma a másodlagos regiszterblokk hiányában csak nehézkesen és nagy overhead árán lenne kivédhető a regisztertartalom rendszeres memóriába mentésével.

- **Utasítás cache**

A Harvard-architektúra nyújtotta előnyök ezen egység segítségével használhatók ki maximálisan. Az utasítás cache-nek köszönhető, hogy utasítások ciklikus végrehajtása esetén három párhuzamos adatmozgatás és két számítási művelet hajtódik végre egy órajelcikluson belül. Ezek részletesebben: Az adatmemória és a kódmemória buszain egy-egy adatot olvasunk be regiszterekbe. (A kódterületen is lehet adatot tárolni.) A szorzó és az összeadó egyidejűleg hajt végre egy-egy műveletet regiszterekben tárolt adatokon. A következő végrehajtandó utasítás kódját pedig az utasítás cache szolgáltatja.

- **Adatcímző egységek**

Az adatcímző egységek egyike a kódterülethez, a másik az adatterülethez tartozik. Ezen egységek hardver szinten támogatják cirkuláris bufferek megvalósítását. Egy egyszerű memóriatömb azáltal lesz cirkuláris buffer, hogy a bufferen belüli címeket az adatcímző egység speciális módon számítja ki. Ez úgy történik, hogy a cirkuláris bufferhez egy mutató regisztert rendel hozzá, amely segítségével a címzés történik. A mutató regiszter a cirkuláris buffer báziscíméhez képesti eltolást tárolja, értéke maximálisan a buffer hossza lehet. Amennyiben a mutató regiszter olyan értékre módosulna, amely nagyobb, mint a buffer hossza, akkor a regiszter tartalma ezen értéknek a buffer hosszával osztott törtrésze lesz. Ilyen módon egy körkörös vagy cirkuláris buffer jön létre, amely nagyon kedvező néhány tipikus jelfeldolgozási algoritmus (FFT, FIR szűrés) megvalósításakor.

- **SRAM memória**

A token belül (vagy on-chip) megvalósított 1 Mbytes SRAM memória miatt nem kell külön külső memóriát biztosítani a processzor működéséhez. Természetesen amennyiben nem lenne elegendő ennyi memória, lehetőség van további külső memória illesztésére. Az SRAM memóriaterület a Harvard-architektúrának megfelelően két részre van osztva, adat- illetve kódterületre. Ez a hagyományos elrendezés, de nincs semmiféle megkötés abból a szempontból, hogy milyen típusú adatokat tárolunk az egyes területeken. Az optimális működési sebesség elérésének érdekében mindig az adott alkalmazás szempontjából mérlegelni kell, hogy milyen elrendezéssel érhető el a lehető legtöbb párhuzamos memóriaművelet. Például FIR szűrés esetén a bemeneti adatokat és a szűrőegységkimeneteket külön memóriaterületen érdemes tárolni.

- **Külső memória és periféria interfész**

Külső memória illesztésére a külső buszon (external bus) keresztül van lehetőség. Erre a külső buszra a belső buszok multiplexálva csatlakoznak, tehát egy külső eszköz mind adat, mind kód memóriaterületre illeszthető. Ilyen módon a DSP-hez külső memórián kívül tetszőleges periféria illeszthető. Ezen az interfészen keresztül csatlakoztatható például FPGA áramkörhöz, vagy többprocesszoros működés esetén (multiprocessing) másik DSP-hez.

- **Host interfész**

A host interfész nyújt lehetőséget szabványos mikroprocesszoros buszokhoz való illesztésére. Ezen az interfészen keresztül a DSP egyszerűen csatlakoztatható személyi számítógéphez.

- **DMA vezérlő**

A DMA (Direct Memory Access) vezérlő nagy tömegű blokkos adatmozgatások esetén használatos. Ekkor a processzor magjának igénybevétele nélkül a DMA vezérlő nagy sebességű adatmozgatást végez a belső SRAM memória és valamely más eszköz között. Ezen eszközök a következők lehetnek: külső memória, külső periféria, host processzor, soros port. Ezen kívül DMA kapcsolat létesíthető a külső memória és más külső eszközök között is

- **Soros portok**

A DSP két szinkron soros porttal rendelkezik, amelynek segítségével különböző digitális perifériákhoz való csatlakozásra nyílik lehetőség. Ezen soros port sebessége széles tartományban mozoghat, maximálisan 40 Mbit/sec.

- **JTAG interfész**

Az IEEE JTAG 1149.1 szabványnak megfelelő port a DSP tesztelésére szolgál. Ezen port segítségével emulálásra van lehetőség, ami azt jelenti, hogy az integrált áramkörön belül vannak megvalósítva az emulátor funkciók. Tehát megfelelő eszköz vagy alkalmazás segítségével e porton keresztül a DSP működése nyomon követhető és tesztelhető. Ezt a portot használja az EZ-ICE Emulátor is.

### 6.2.3. Értékelés

Kijelenthető, hogy a processzor felépítése és különböző funkciói kielégítők abból a szempontból, hogy a megvalósítani kívánt jelfeldolgozási algoritmusok széles körének implementálását hardver szinten támogatja. Még a legmagasabb mintavételi frekvencia mellett is csatornánként ( kb. 830 utasítás ) hajtható végre, amely az overhead miatt egy kb. 800 együtthatós FIR szűrő megvalósítására elegendő. Ezért az átlagos alkalmazások esetén elegendően hosszú jelfeldolgozási rutinok futtatására van lehetőség, ami biztosítja a problémamentes működést.

### 6.3. A VisualDSP++ fejlesztői környezet

A DSP programok fejlesztése az EZ-KIT kártyával nagymértékben együttműködni képes VisualDSP++ fejlesztői környezet segítségével történt. A VisualDSP++ az Analog Devices cég szoftver terméke. Ez egy személyi számítógépen futó integrált fejlesztői környezet, amely azt a célt szolgálja, hogy az AD DSP-in futó programokat fejlesszék ki segítségével. Az alkalmazás Windows grafikus környezetben fut.

A fejlesztői környezet két független alkalmazásból áll, a Projekt szerkesztőből és a Debugger-ből.

- **Projekt szerkesztő**

A Projekt szerkesztőben projekt fájlokat lehet megnyitni és szerkeszteni. Egy ilyen projekt tulajdonképpen egymáshoz rendelt forráskód fájlokat jelent, amelyekből az alkalmazás egy DSP-re letölthető fájl generál.

A kimeneti fájl típusa beállítható a Projekt tulajdonságok ablakban. A lehetőségek: debuggolható verzió, kész verzió, vagy EPROM-ba égethető bit-file. A kimeneti fájl

kiterjesztése ettől a beállítástól függ, A kész verzió kiterjesztése .exe, a debuggolható verzióé .dxe. A forráskód fájlok lehetnek gépi nyelven megírtak (kiterjesztésük: .asm), C nyelven megírtak (.c), vagy a linkelést leíró úgynevezett Linker Description File-ok. (.ldf).

A Projekt szerkesztőn belül a következő funkciók vannak megvalósítva:

- **C fordító**

E funkció biztosítja a DSP-k C nyelvű programozását, amely sok esetben kényelmesebb a nehézkesnek tűnő gépi kóddal szemben. Használata mégsem igazán elterjedt, mert a C fordító nem tudja maximálisan kihasználni a DSP hardver adta lehetőségeket. A DSP-eket pedig éppen azért használják a különböző időkritikus jelfeldolgozó algoritmusok megvalósítására, mert hardver szinten támogatják a jelfeldolgozási műveletek végrehajtását. Ezeket a speciális hardver lehetőségeket viszont csak egy assembly nyelven megírt program tudja igazán kihasználni. Sok esetben használatos az az egészséges kompromisszumot jelentő megoldás, hogy a DSP programok sokszor lefutásra kerülő részleteit gépi nyelven írja meg a fejlesztő, míg az egyszer lefutó inicializáló jellegű programrészleteket C-ben. Így a futási idő túlnyomó részében futó rutinok assembly nyelven vannak megírva, tehát a program futása nem lassul le lényegesen.

- **Compiler**

A compiler végzi a gépi kódban megírt forrásfájlok valamint a C fordító által lefordított fájlok szintaktikai és szemantikai ellenőrzését.

- **Linker**

A linker felelős azért, hogy az egy projekten belüli különálló fájlokat egybefűzze, és előállítsa a kimenet .exe vagy .dxe fájlt. Minden projekthez tartozik egy Linker Description File (LDF) is, amely szintén a projekt részét képezi. Amennyiben a felhasználó nem ad meg ilyen LDF-et, akkor a Projekt szerkesztő alapértelmezésként egy gyári LDF-et használ.

A Linker Description File a linker számára tartalmaz lényeges információkat, tehát nem közvetlenül a DSP-n futó program része. Az LDF egy egyszerű, néhány utasításból álló nyelven megírt leírófájl, amely alapján a linker összefűzi a forrásfájlok kód és adatrészleteit. Ez a leírófájl tartalmazza azt a lényeges információt, hogy a különböző forrásfájlok adat- és kódrészleteit a DSP-n fizikailag melyik memóriaterületre kívánja letölteni a felhasználó.

- **Loader és Splitter**

Ez a két funkció akkor használatos, ha a kész DSP programot EPROM-ba kívánjuk égetni. Ezen alkalmazások segítségével állíthatók elő a szükséges bit fájlok.

- **Debugger**

A fejlesztői környezet másik nagy részegysége a Debugger alkalmazás. Ez az alkalmazás lehetőség biztosít a megírt DSP programok kipróbálására futás közben. Lehetőség van a program futásának részletes vizsgálatára, töréspontok helyezhetők el, lehet léptetve végrehajtani az utasításokat, a memóriaterületek, valamint a rendszer regisztereinek tartalma folyamatosan monitorozható. A Debugger „bementi adata” a Projekt szerkesztő által előállított .dxe kiterjesztésű fájl. A DSP program futtatásának helye beállítható, ez lehet emulátor vagy szimulátor.

- **Szimulátor**

Ebben az esetben a program fizikailag nem töltődik le a DSP-re, hanem az alkalmazás a személyi számítógépen szimulálja a DSP működését. Ez a szimuláció minden részletre kiterjed, tehát a DSP majdani várható működése vizsgálható. Ilyen módon azonban nem lehet

kipróbálni sem a fizikai DSP alapú rendszer többi egységét, illetve az egységek összehangolt működését sem. Ezért, ha lehetőség van rá, akkor az emulátor használata előnyt élvez.

#### ▪ **Emulátor**

Az emulátor használatára akkor van lehetőség, ha rendelkezésre áll az ADI által gyártott EZ-ICE emulátor kártya a személyi számítógépben. Ehhez tartozik egy csatlakozó, amelyet az ADI fejlesztői kártyáin található JTAG Emulátor csatlakozóval lehet összekötni. Erre a processzor belső (vagy on-chip) emulálását lehetővé tevő szabványos JTAG portjának lábai vannak kivezetve. Az emulátor ezen az interfészen keresztül vezérli a teljes vizsgálatot. Ugyanitt történik a Projekt szerkesztő által előállított program letöltése a DSP memóriájába. A letöltés után a processzort a Debugger Halt állapotban tartja. Ekkor lehetőség van a program futtatására, az utasítások léptetett elvégzésére vagy töréspontok elhelyezésére. A DSP összes regiszterének tartalma lekérdezhető és igény szerint módosítható. A memóriaterületek teljes területének tartalma fájlba írható (dump) vagy feltölthető (fill). A fenti műveletek elvégzésének mindegyikét a JTAG szabvány szerint a DSP hardveresen támogatja, ezért elvégzésükhöz nincs szükség külön monitorprogram letöltésére. A Debugger alkalmazás segítségével a program futásának teljes körű vizsgálata lehetővé válik.

Ha rendelkezésre áll az EZ-KIT Lite fejlesztői kártya, akkor emulálásra lehetőség van soros porton keresztül is, ugyanis a kártyán futó keretprogram – amit a boot szekvencia után az EPROM-ból tölt be – ezt lehetővé teszi. A lehetőségek ugyanazok, mint JTAG emulátor használata esetén, a különbség csak az, hogy az RS-232 kommunikációhoz szükséges kernel programnak a memóriában kell lennie.

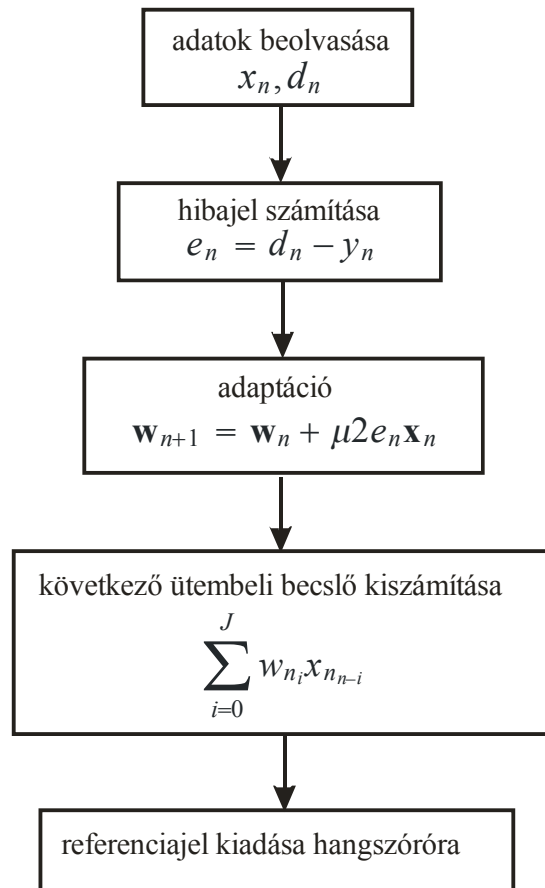
## 7. Megvalósított zajcsökkentő eljárások

A zajcsökkentő algoritmusokat az ADSP 21061-es EZ Kit Lite kártyán valósítottam meg. Az aktív zajcsökkentő eljárások, ahogyan a bevezetőben is említettem 1-2 kHz-ig működnek hatékonyan. Az algoritmusok számítási igényét is figyelembe véve  $f_s = 2$  kHz-et választottam mintavételi frekvenciának. Mivel a kártyán lévő AD-átalakító nem állítható 8 kHz-nél kisebb mintavételi frekvenciára, ezért négyszeres decimálást hajtottam végre. Azért, hogy a jelfeldolgozó algoritmusok végrehajtására elég idő álljon a rendelkezésre, a jelfeldolgozást nem a timer megszakítás rutinban, hanem a főprogramban végeztem. A timer megszakítás rutinban csupán a bejövő adatok: referenciajel, hibamikrofonok jeleinek beolvasását és a kimenő adatok: beavatkozó hangszórók jeleinek kiírását végeztük.

### 7.1. Átviteli függvény mérése LMS algoritmussal

Ahogy azt a 3.1.1. alfejezetben bemutattam, az LMS alapú zajcsökkentő rendszerek megvalósításakor a másodlagos átviteli út transzfer függvényét LMS algoritmussal mértem. Ez azt jelentette, hogy a DSP kártya bemenetére egy zajgenerátorból szélessávú zajt (1 kHz-ig) adtam. Ez volt a 3.2. ábrán  $x_n$ -el jelölt referenciajel. Ahhoz, hogy az akusztikus átvitelt mérjem a referenciajelet változatlan formában ki lett adva a DSP kártya kimenetén, ami a beavatkozó hangszóróra volt kötve. A megkívánt jel pedig a hibamikrofon jele volt. A referenciajelet tehát a beavatkozó hangszórón adtam ki, és a megkívánt jel a hibamikrofon jele volt, azaz az akusztikus átvitel bemenetét a referenciajellel gerjesztettem, a hibamikrofon jelén keresztül pedig az akusztikus rendszer kimenetének jelét vettem, ami ebben az esetben a megkívánt jel. Ezzel az eljárással a 3.2. ábrán látható elrendezést valósítottam meg.

Az átviteli függvény-mérő program felépítése a 7.1. ábrán látható:



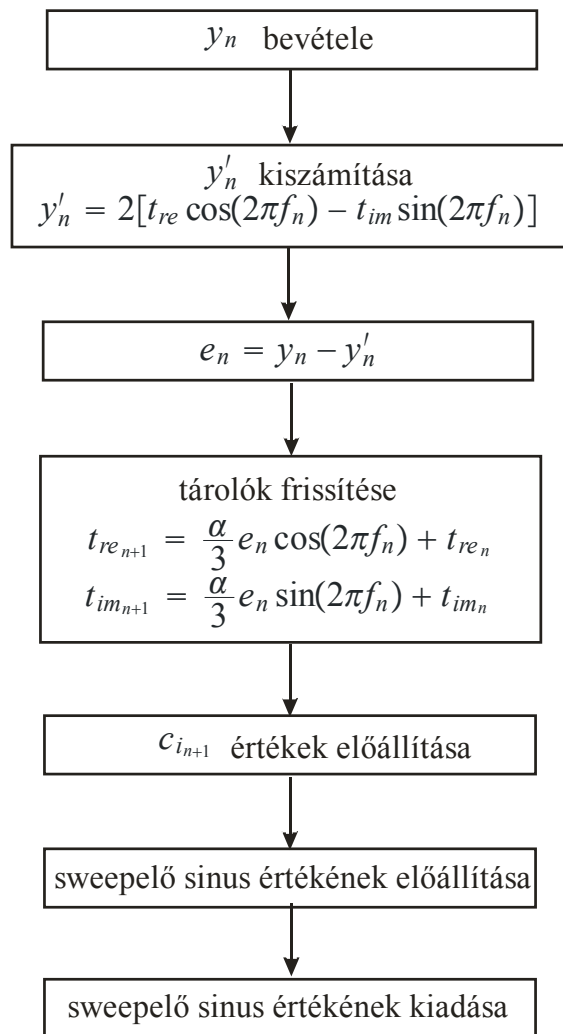
7.1. ábra. Átviteli függvény mérése LMS algoritmussal

## 7.2. Átviteli függvény mérése rezonátoros struktúrával

Az átviteli függvény-mérő program a 4.3.3. fejezetben leírtakkal összhangban két részből áll: az egyik rész diszkrét időközönként frekvenciát váltó szinuszos gerjesztőjelet állít elő a kimenet számára (azaz a szinuszjel frekvenciája bizonyos időközönként ugrásszerűen változik), a másik rész pedig egy AC rezonátorpár és egy DC rezonátor segítségével az identifikálást végzi. Mivel a rezonátorok komplex jelekkel dolgoznak, ezért minden egyes rezonátor megvalósításakor két regisztert kellett felhasználnom (a DSP nem támogatja a komplex jeleket) egyet a komplex szám valós részének ( $t_{re}$ ), egyet a képzetes résznek ( $t_{im}$ ).

Mivel valós jeleket dolgozunk fel, ezért a rezonátoros struktúra megvalósításánál a következőképpen járhatunk el. Az AC rezonátorpár helyett egy darab AC rezonátort valósítottunk meg, és a hibajel kiszámításakor ezen rezonátor tárolója valós részének a kétszeresével számoltunk. Ezt azért tehetjük meg, mivel valós jelek esetén a pozitív frekvencián lévő jel a negatív frekvencián lévő jel konjugáltja, így összegük a valós rész kétszerese lesz. A program folyamatábrája a 7.2. ábrán látható.



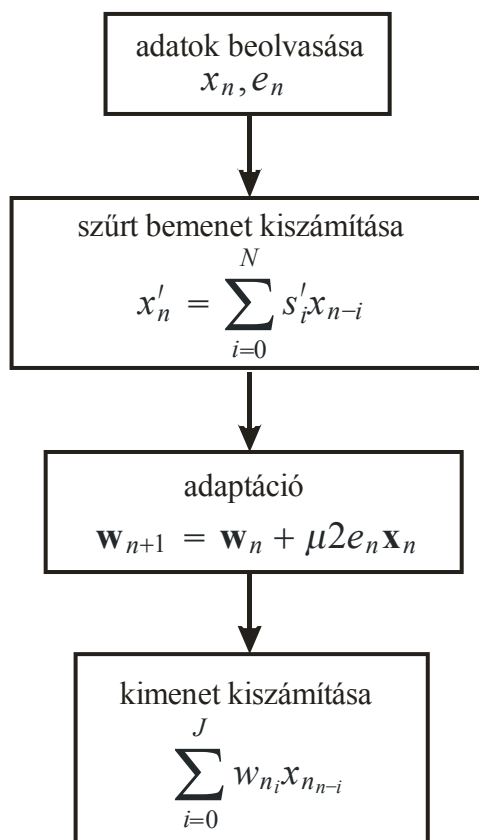


7.2. ábra. Átviteli függvény mérése rezonátoros struktúrával

### 7.3. *Filtered-X LMS algoritmus*

A Filtered-X LMS algoritmust egy off-line identifikáció előzi meg, hiszen a eljárás működéséhez szükség van a másodlagos transzfer függvény ismeretére. Ennek az identifikációnak a megvalósítását mutattam be a 7.1. alfejezetben. Az identifikáció után kapott együtthatókészlet a 3.9. ábrán  $S'(z)$ -vel jelölt FIR szűrő.

Az eljárás az elnyomandó zajból vett referencijellel dolgozik, ezt a zajforrás mellé elhelyezett mikrofonnal vettem. Az algoritmus másik bemenőjele egy másik mikrofon, a hibamikrofon jele. E két jel alapján a program előállítja a beavatkozó hangszóró jelét, ami a zajcsökkentést biztosítja. A megvalósított program blokkvázlatát a 7.3. ábra mutatja:

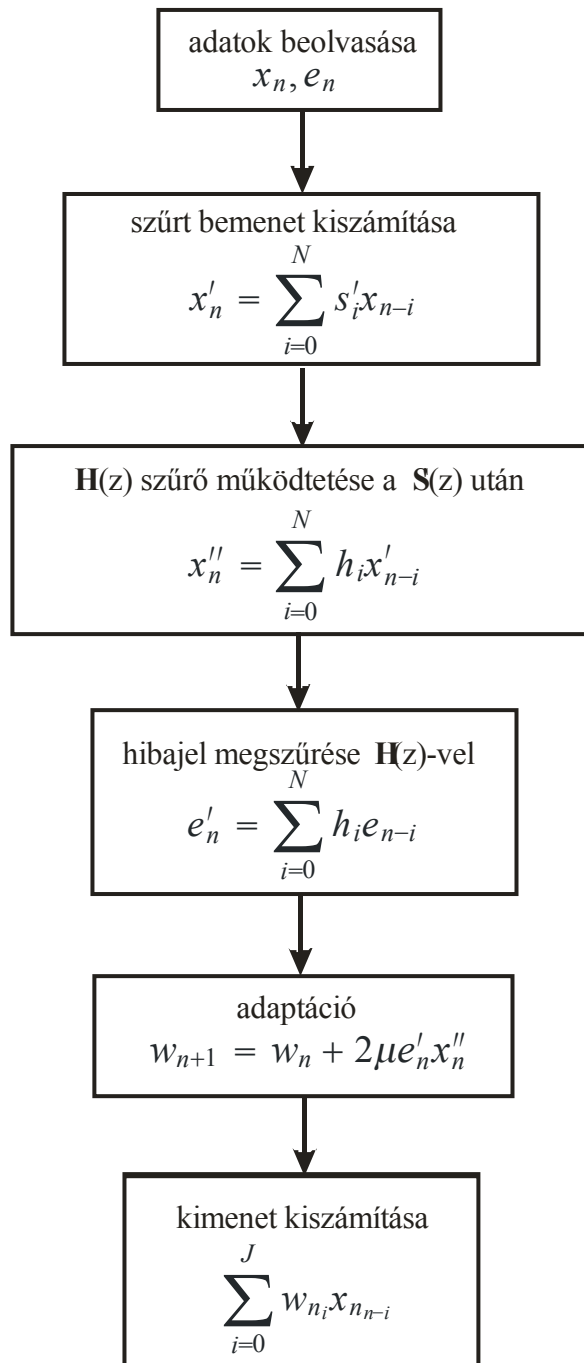


7.3. ábra. A Filtered-X LMS algoritmus működése

#### 7.4. *Filtered-EX LMS algoritmus*

Ezt az eljárást, akárcsak az előző alfejezetben bemutatottat, off-line identifikáció előzi meg. Itt is a 7.1. alfejezetben bemutatott programot használtam, majd az így kapott együttthatókészlet segítségével MATLAB-ban frekvenciamintavételező eljárással előállítottam az algoritmus működéséhez szükséges, a 3.17. ábrán bemutatott  $H(z)$  szűrőt, ami a másodlagos úttal és modelljével tulajdonképpen kaszkád kapcsolatban működve biztosítja az eredően egységnyi körül ingadozó átvitelt.

A megvalósított eljárás a Filtered-X LMS algoritmushoz hasonlóan a zajforrásból vett referenciajelet és a hibajelet igényli a működéshez. Ezeket a 7.3. alfejezetben leírtakkal megegyezően biztosítottam. A DSP program folyamatábrája a 7.4. ábrán látható:

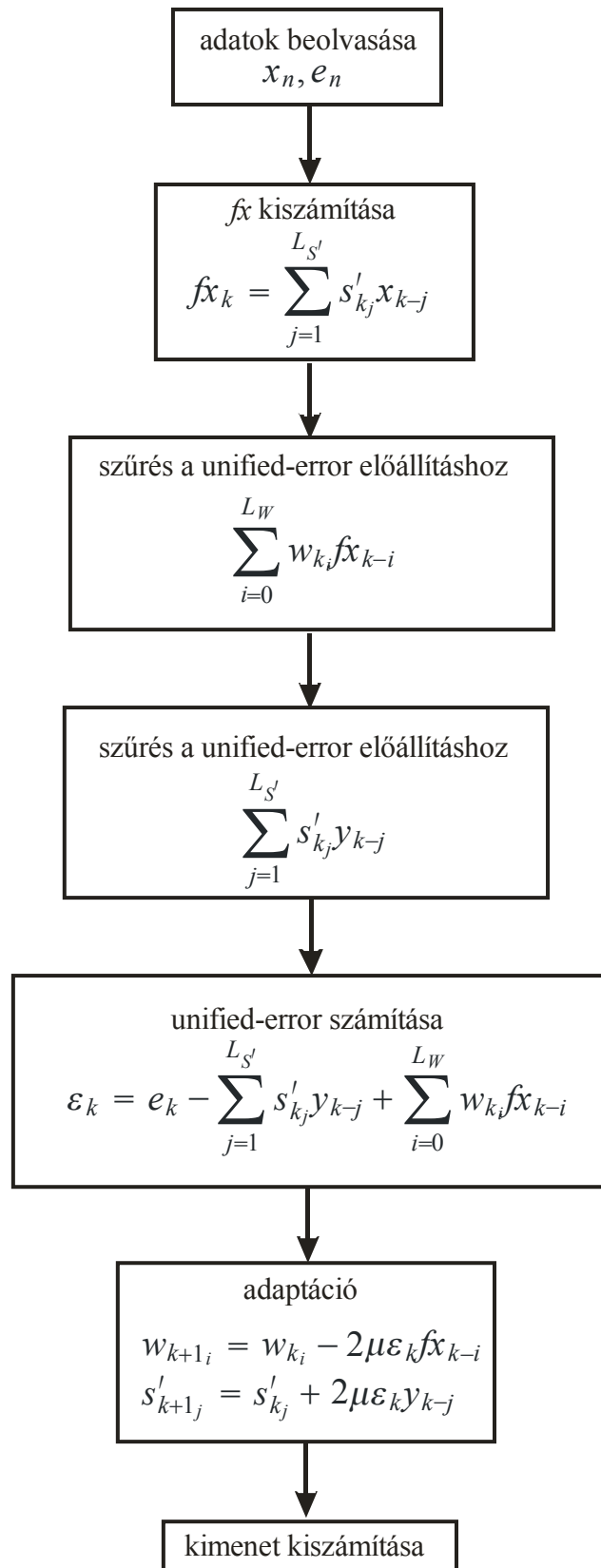


7.4. ábra. A Filtered-EX LMS algoritmus blokkvázlata

## **7.5. *Unified-error Filtered-X LMS algoritmus***

A Unified-error Filtered-X LMS algoritmus működéséhez elvileg nem szükséges off-line identifikáció, a tapasztalat azonban azt mutatta, hogy az 5.7. ábrán látható  $S'(z)$  szűrő együtthatóit nulláról indítva az algoritmus instabil is lehet. Ezért a megvalósított eljárásban a program inicializáló részében az előbbi szűrőegyütthatóknak egy korábban LMS algoritmussal identifikált átviteli függvény értékeit adtam. Ezek az együtthatók nem egyeztek meg az aktuális környezet átvitelével, csak inicializálásként való felhasználásuk volt fontos.

Az eljárás a Filtered-X LMS és Filtered-EX LMS algoritmushoz hasonló bemeneti jeleket használ, és kimenet szintén a beavatkozó hangszórót vezérli. Az eljárás folyamatábráját a 7.5. ábra szemlélteti.

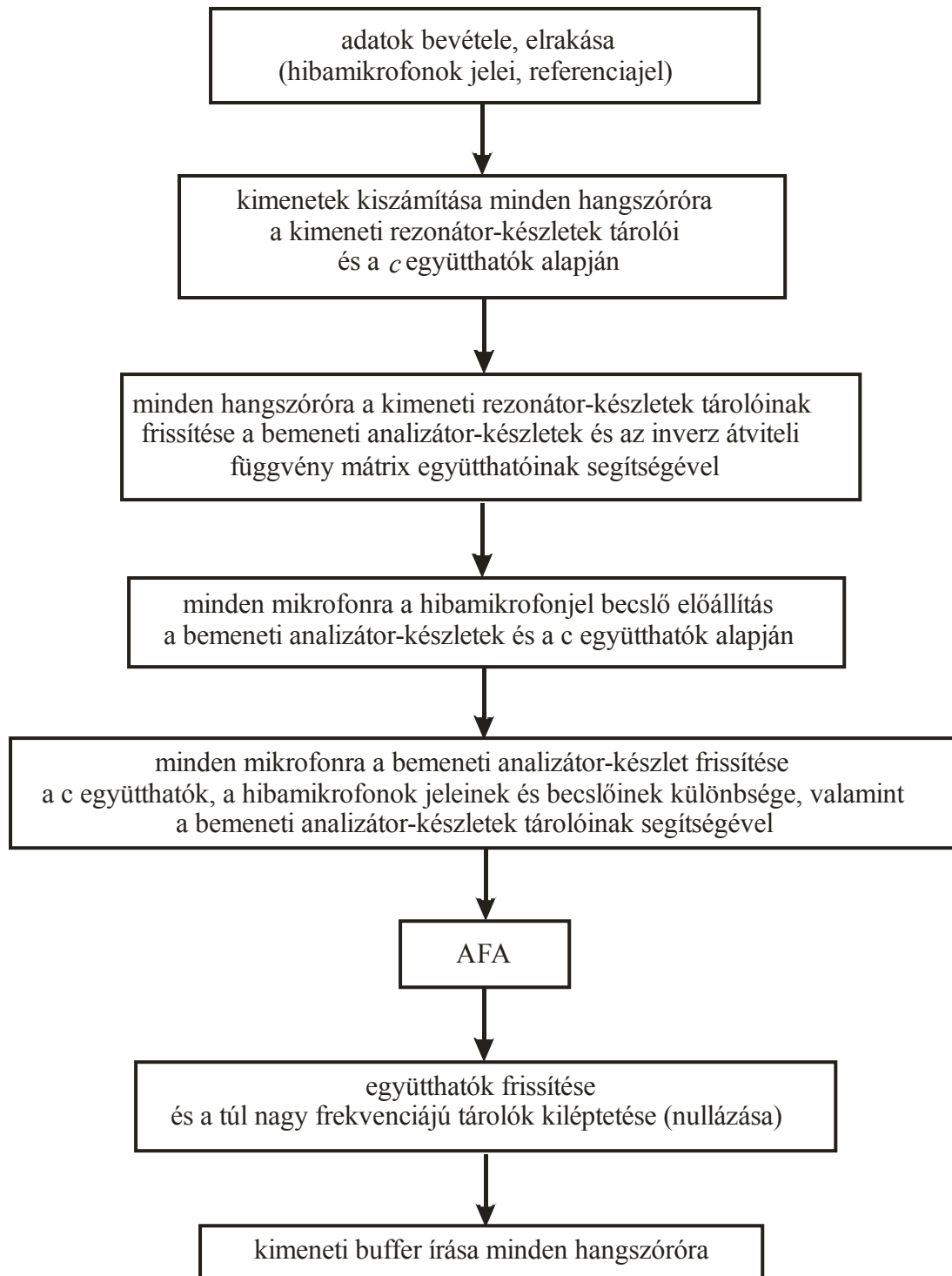


7.5. ábra. A Unified-error Filtered-X LMS algoritmus folyamatábrája

## 7.6. Rezonátoros zajcsökkentő struktúra

A rezonátoros zajcsökkentő programot úgy valósítottam meg, ahogy az a 4.12. ábrán látható. Azaz egy bemeneti analizátorkészlet, egy kimeneti rezonátorkészlet és egy ezektől független, az AFA algoritmus számára szükséges rezonátorkészletet használtam. Az eljárás úgy lett megírva, hogy néhány egyszerű paraméter módosításával  $n*m$ -es zajcsökkentő struktúra nyerhető, ahol  $n$  a hibamikrofonok,  $m$  pedig a beavatkozó hangszórók száma. Ezzel a 4.13. ábrán látható struktúra kapható. Méréseket egy csatornás esetre végeztem, ami tulajdonképpen a több csatornás zajcsökkentő algoritmus egy speciális esete ( $n=m=1$ ), mert így az eljárás összehasonlítható a többi megvalósítással.

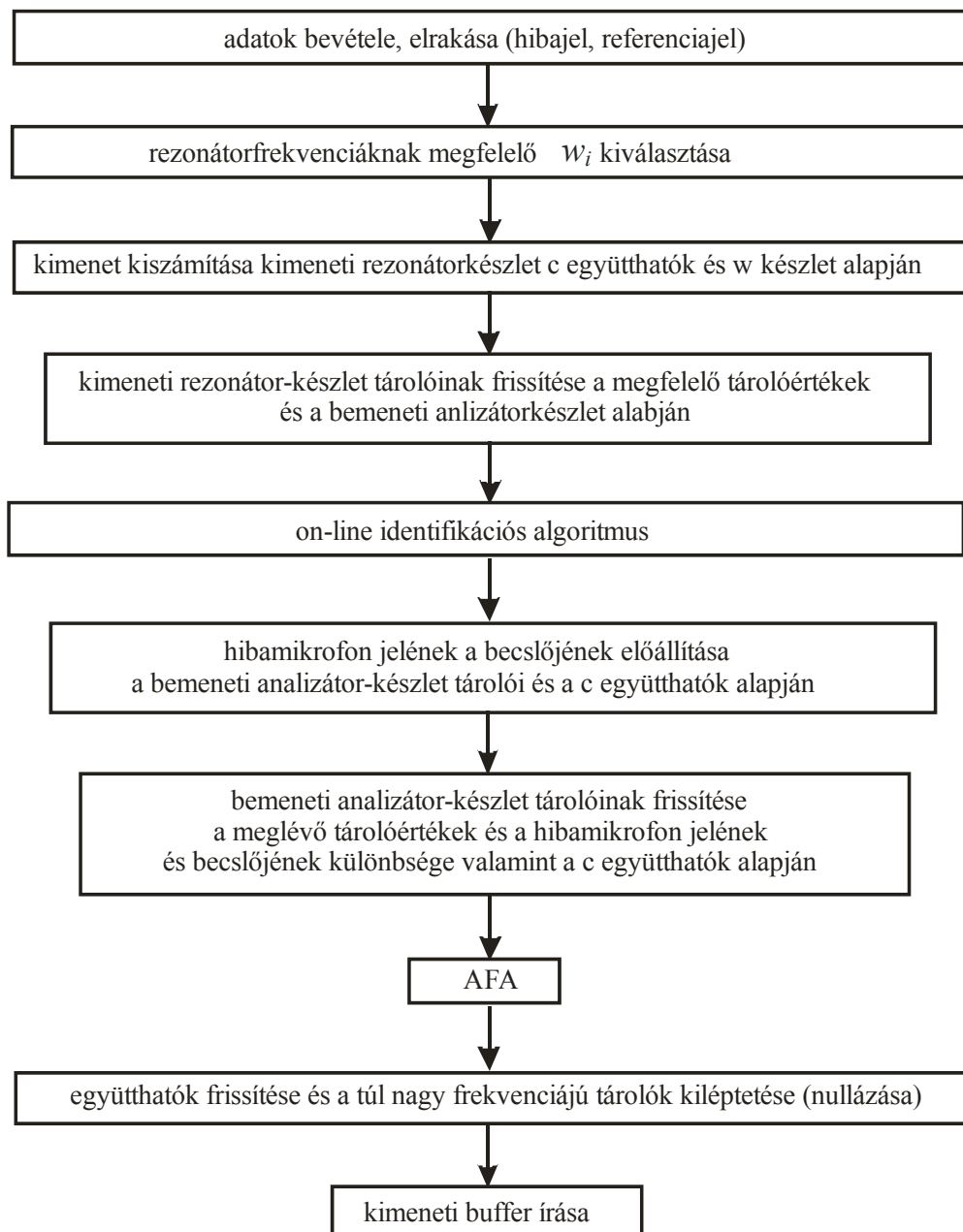
Az AFA algoritmushoz szükségem volt egy referenciajelre a zajforrásból. A programot úgy lett megkonstruálva, hogy az AFA algoritmus a rezonátorfrekvenciákat a  $c_{n,k}$  együtthatók változtatásán keresztül állította. A zajcsökkentő struktúra megvalósításánál a jelmodell alapú megfigyelő integrátoros modelljét használtam. A program folyamatábráját a 7.6. ábra szemlélteti.



7.6. ábra. Rezonátoros zajcsökkentő program folyamatábrája

## 7.7. Rezonátoros zajcsökkentő struktúra on-line identifikációval

Az eljárás megvalósítása során a 4.12. ábrán látható megnövelt konvergenciasebességű struktúrát használtam, amibe beágyaztam az on-line identifikációs algoritmust. Az algoritmus megírása során kihasználtam azt, hogy egycsatornás esetben a kimeneti rezonátorkészlet tárolói abszolút értékének deriváltja a bemeneti analizátorkészlet tárolóinak abszolút értéke. A program folyamatábráját a 7.6. ábra szemlélteti.



7.6. ábra. Rezonátoros struktúra on-line identifikációval



## 8. Teszteredmények

Ebben a fejezetben bemutatom a mérési elrendezést, amelyben teszteltem a megvalósított algoritmusokat, majd áttekintem az általam végzett mérések eredményeit, melynek alapján összehasonlítom a különböző eljárásokat, rámutatva előnyeikre és hátrányaikra.

Az LMS alapú algoritmusok közül három eljárást valósítottam meg: a Filtered-X, a Filtered-EX és a Unified-error Filtered-X LMS algoritmusokat. Ezen algoritmusok működéséhez szükség volt a másodlagos út off-line identifikációjára. A Unified-error algoritmusnál ezeknek az együtthatóknak csak az  $S'(z)$  szűrő inicializálásában van szerepük ahogyan azt az 5.2. fejezetben említettem, az algoritmus további működéséhez nincs szükség a másodlagos átviteli út további modelljeire, on-line követi annak változásait. Az off-line identifikáció elvégzésére a 7.1. alfejezetben leírt struktúrát használtam.

A megvalósított eljárásokat több elnyomandó jellel is teszteltem. Az LMS alapú algoritmusokat a következő jelekkel:

- szélessávú fehér zaj (200 és 800 Hz között)
- keskeny sávú fehér zaj (450 és 550 Hz között)
- háromszögjel (alapharmonikus frekvenciája 150 Hz)

Ahhoz, hogy bemutassam az általam megvalósításra választott LMS algoritmus alapú on-line identifikációs zajcsökkentő algoritmus, a Unified-error Filtered.X LMS eljárás működését, két összehasonlító mérést végeztem az előbbi eljárás és a Filtered-EX algoritmus között, melynek során az egyik esetben az elnyomandó jel keskeny sávú fehér zaj volt (450 és 550 Hz között), a másik esetben 150 Hz alapharmonikus frekvenciájú háromszögjel. A mérés lényege az volt, hogy a hibamikrofont körülbelül fél méterrel elmozdítottam eredeti helyzetéből, ezzel megváltoztatva a másodlagos átviteli út transzfer függvényét. Az eredmény: a unified-error algoritmus a mikrofonnak ebben a pozíciójában is biztosítani tudta az átvitelt, míg a Filtered-EX LMS algoritmus instabillá vált.

A rezonátoros struktúrákkal az összehasonlíthatóság érdekében 150 Hz alapharmonikus frekvenciájú háromszögjelre végeztem méréseket. Ezen kívül az on-line identifikációs rezonátoros struktúra működésének igazolásához ebben az esetben is elvégeztem a másodlagos átviteli út változtatását, az előző bekezdésben leírtak szerint. A hibamikrofont ebben az esetben is ugyanabból a pozícióból mozgattam ugyanabba a pontba, mint az LMS alapú algoritmusok tesztelésekor. A rezonátoros struktúra elnyomását a fentiekén kívül teszteltem sweepelő szinuszos elnyomandó jelre is. A mérés alapján megfigyelhető, hogy a rezonátoros zajcsökkentő rendszer az adaptív Fourier-analízátor struktúrával kiegészülve hogyan képes követni az elnyomandó jel frekvenciájának változását.

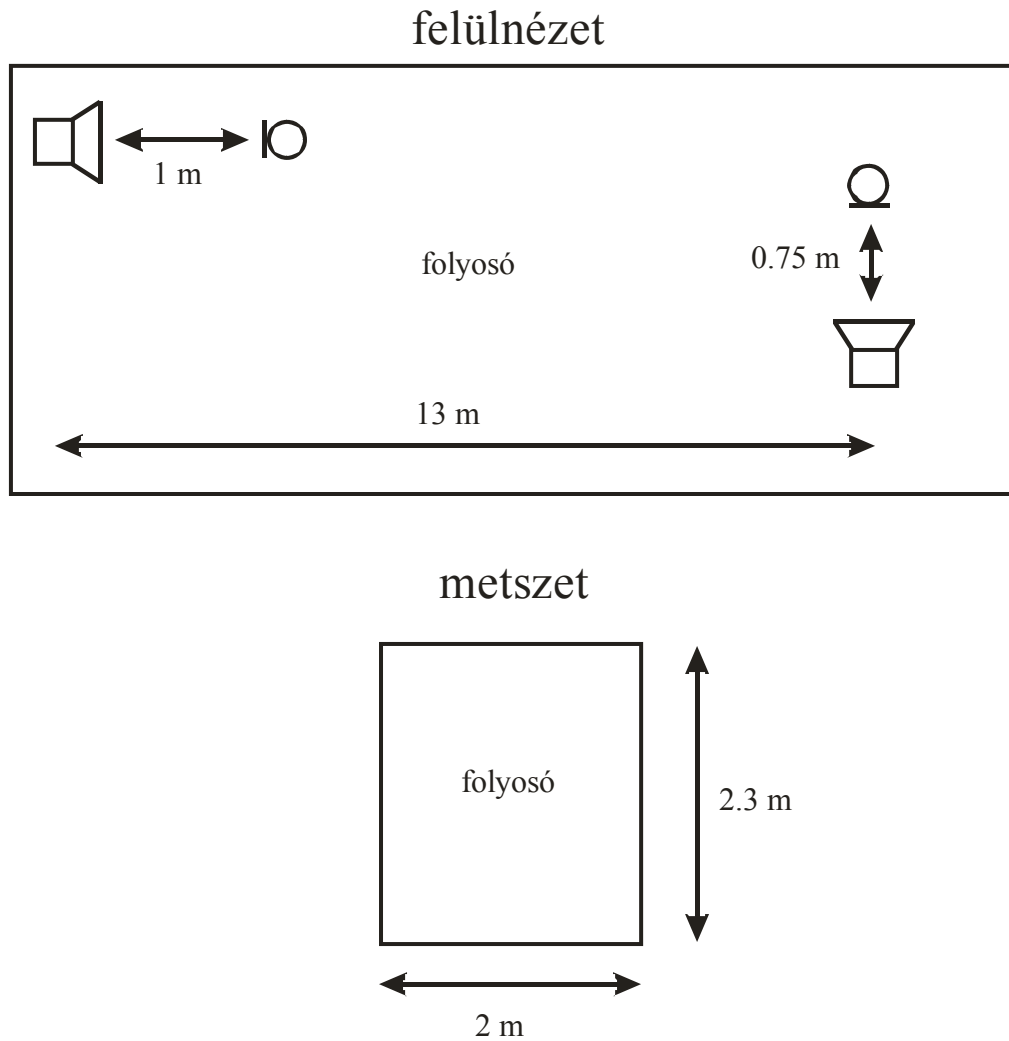
## 8.1. Mérési összeállítás

A méréseket olyan elrendezésben végeztem, amely valós helyzetet modellez. A modellezés ebben az esetben azt jelenti, hogy az elnyomandó zajt én generáltam. A mérések helyszíne egy 2.3 m magas és 2 m széles, az egyetem I épületének 3. emeletén található folyosó volt. Mivel egycsatornás esetre végeztem méréseket, ezért összesen két hangszóróra volt szükség. Az egyik generálta az elnyomandó jelet, a másik pedig a beavatkozó hangszóró volt. A két hangszórón kívül két mikrofont használtam. Az egyik a hibamikrofon volt. A másik mikrofonra azért volt szükség, mivel fehér zaj elnyomásakor a referenciajel vételét csak akusztikusan, azaz mikrofon segítségével lehet megoldani, ezen kívül az LMS algoritmus alapú eljárásoknak a referenciajelben a teljes elnyomandó jelet meg kell kapnia, még periodikus zaj elnyomásánál sem elég csupán az egyik harmónikus. Ezért ezt a mikrofont a zajforrástól (ebben az esetben egy hangszóró) kis távolságra, 1 m-re, a zajforrással szemben helyeztem el. A rezonátoros struktúra használatakor a referenciajelet elektronikus úton vettem. Ez azért tehető meg, mivel a rezonátoros struktúra rezonátorpozícióit egy adaptív Fourier-analizátor struktúra hangolja, aminek azonban csak a periodikus jel alapharmónikusára van szüksége, ami a gyakorlatban valamilyen közvetett úton elektronikus formában megszerezhető (gondoljunk motorzaj elnyomásánál a motor fordulatszámának rezgésérzékelővel történő meghatározására).

A másodlagos út identifikációjakor megfigyelhető, hogy az identifikáció nem csupán az akusztikus útra terjed ki, hanem tartalmazza például a DSP kártya kimenetén levő DA, valamint a hibamikrofon felőli bemeneten levő AD átalakító késleltetését is. Azért emeltem ki ezt a két eszközt, mivel késleltetésük nagyban meghatározza a másodlagos út futási idejét. A másodlagos út előbb említett nagy késleltetése miatt az elsődleges útban is nagy késleltetést kellett biztosítani, azért, hogy a rendszer működőképes legyen. Ezért a zajforrást a hibamikrofontól távol, körülbelül 13 m-re helyeztem el. A hibamikrofon és a beavatkozó hangszóró távolsága 0.75 m volt.

A mérést nagy teljesítményű hangfalakkal végeztem, amelyeknek méretei is igen tekintélyesek. A hangfalak négyzetes hasáb alakúak, körülbelül 0.5 m\*0.5 m\*1.1m méretűek. A hangszóró a hangfalakon 0.9 m magasságban található. A két mikrofont két asztalon helyeztem el a fent említett távolságokra a hangszóróktól.

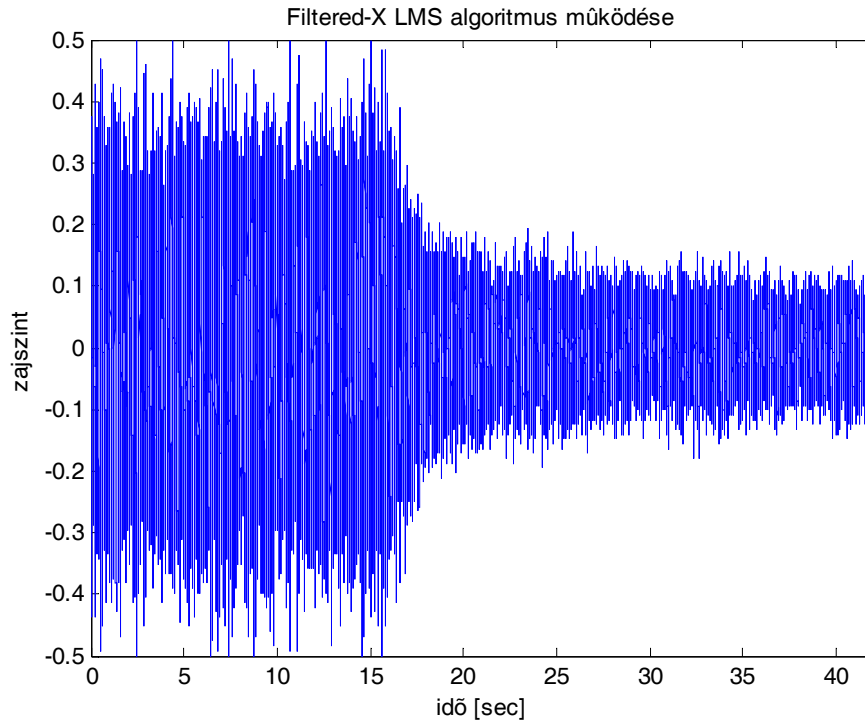
A folyosó méreteit, valamint a mérési elrendezést a 8.1. ábra mutatja (az ábra nem méretarányos).



8.1. ábra. A mérési elrendezés felülnézetből, és a folyosó méretei metszeti ábrán

## 8.2. *Filtered-X LMS algoritmus*

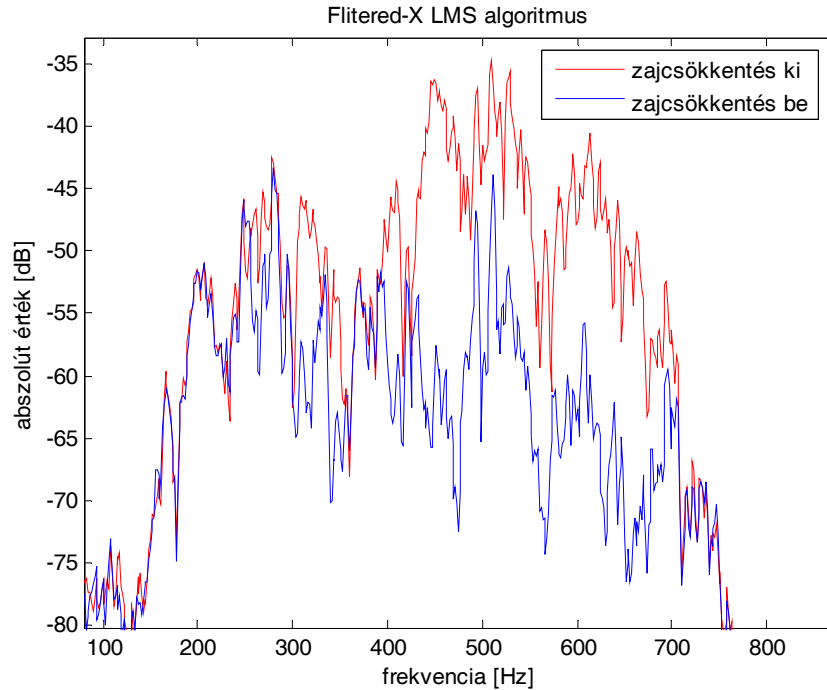
Az első méréssel azt vizsgáltam, hogyan képes az algoritmus a mintavételi frekvenciához képest széles sávú zaj elnyomására. A mintavételi frekvencia 2 kHz volt, a zaj pedig az első esetben 200 és 800 Hz között fehér spektrumú zaj volt, amit egy másik, zajcsökkentésre nem használt DSP kártyán futó programmal generáltam. A zajcsökkentést csak a mérés 15. másodpercében kapcsoltam be, hogy jól megfigyelhető legyen a zajcsökkentés előtti, a zajcsökkentés bekapcsolása utáni állapot, valamint a rendszer beállási ideje. A rendszer időtartománybeli viselkedését mutatja a 8.2. ábra.



8.2. ábra. A Filtered-X LMS algoritmus működése az időtartományban szélessávú zaj esetén, mintavételi frekvencia 2 kHz, a zajcsökkentés 15 s után lett bekapcsolva

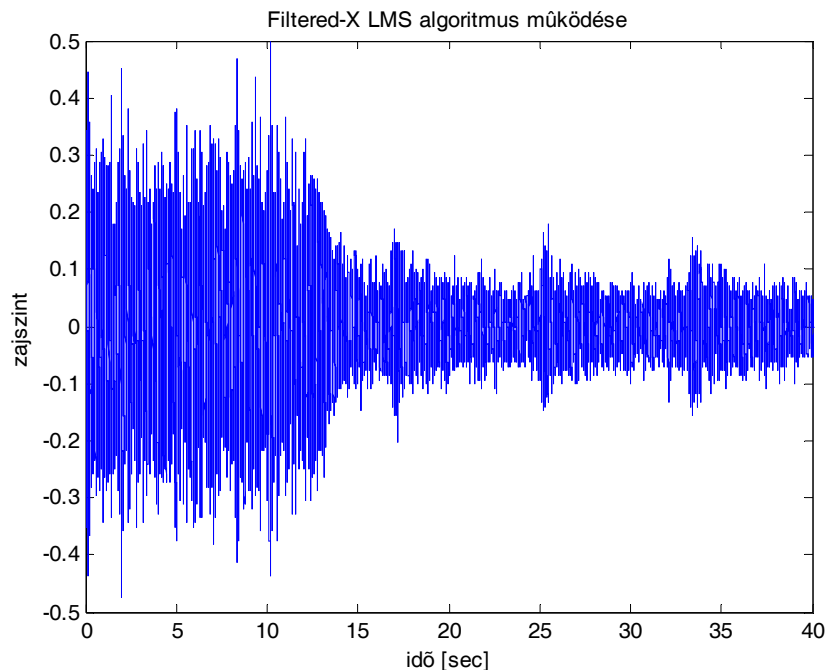
A 8.2. ábrán jól megfigyelhető, hogy a rendszer beállási ideje szélessávú zajra igen nagy, körülbelül 10 s (a 15-től a 25. másodpercig tart). Ez megfelel a vártnak, hiszen a Filtered-X LMS algoritmus fő hátránya a kis konvergenciasebessége.

Ugyanennek a kísérletnek a spektrális képét mutatja a 8.3. ábra. Az ábrán a zajcsökkentés előtti állapotot piros, a zajcsökkentés bekapcsolása utáni állapotot kék színnel jelöltem. A spektrumokat természetesen az állandósult állapotokra jellemző mintákból számoltam.



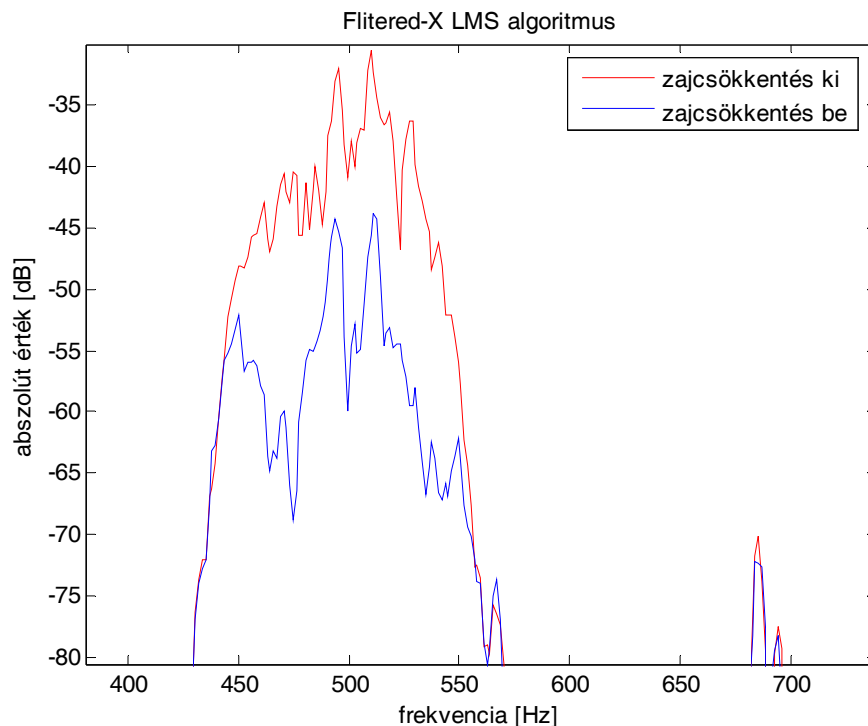
8.3. ábra. A Filtered-X LMS algoritmus működése a frekvenciatartományban szélessávú zaj esetén, mintavételi frekvencia 2 kHz

A következő kísérletben ugyanezt az elrendezést használtam, csak az elnyomandó jel változott. Szélessávú zaj helyett most keskenysávút alkalmaztam, 450 és 550 Hz között fehér spektrumú zaj volt a zavaró jel. Ennek a kísérletnek az időtartománybeli alakulását mutatja a 8.4. ábra. A zajcsökkentés körülbelül a 10. s után lett bekapcsolva.



8.4. ábra. A Filtered-X LMS algoritmus működése az időtartományban keskeny sávú zaj esetén, mintavételi frekvencia 2 kHz, a zajcsökkentés 10 s után lett bekapcsolva

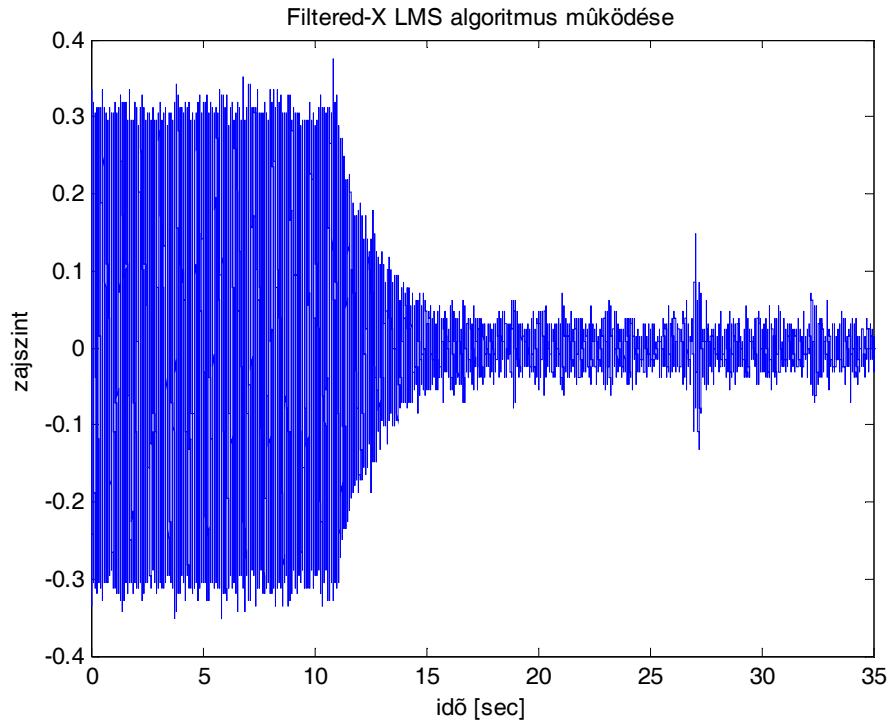
Az ábrára ugyanúgy jellemző a lassú beállítás, ami ebben az esetben is körülbelül 10 s. Ugyanennek a kísérletnek az elnyomást jellemző frekvenciatartománybeli alakulása látható a 8.5. ábrán.



8.5. ábra. A Filtered-X LMS algoritmus működése a frekvenciatartományban keskeny sávú zaj esetén, mintavételi frekvencia 2 kHz

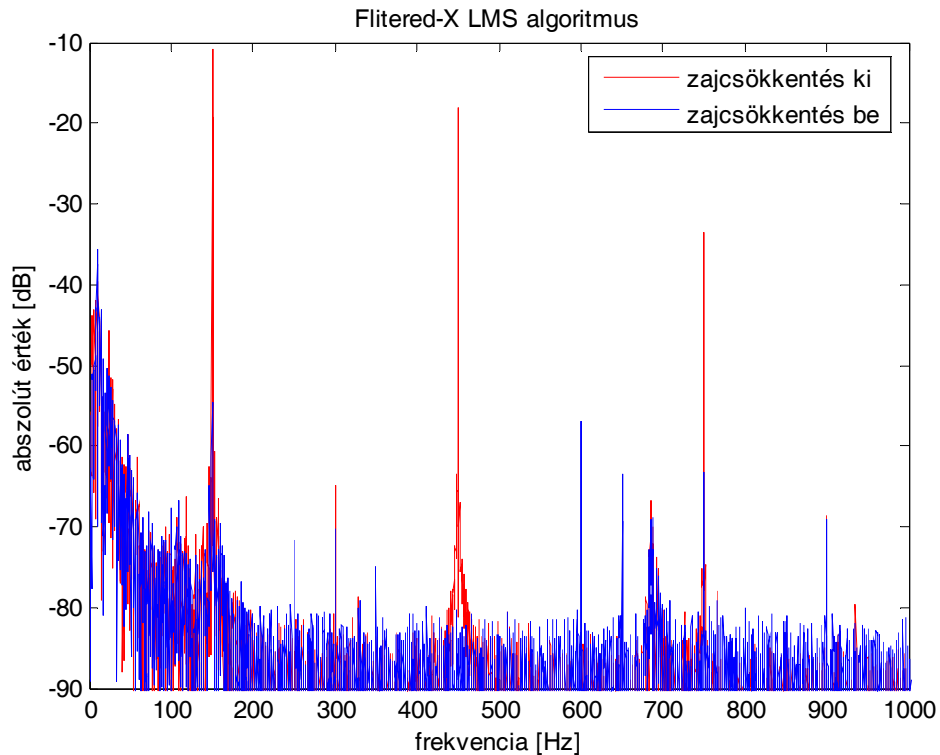
Az ábrán jól látható, hogy a mérést megzavarta egy kis teljesítményű körülbelül 680 Hz középfrekvenciájú zaj (lehet például légkondicionáló, vagy más egyéb gép zaja), ami valószínűleg a referencijellel nem korrelált, vagy csak nagyon kis értéke miatt nem tudta elnyomni az algoritmus. Az elnyomás ezen kívül körülbelül egyenletes minden frekvencián, a zajcsökkentés bekapcsolása utáni spektrum alakja körülbelül követi az elnyomandó zaj spektrumáét.

A harmadik kísérletben az elnyomandó jel egy periodikus jel volt. A 150 Hz alappfrekvenciájú háromszögjellet függvénygenerátorral állítottam elő. A gerjesztés tehát háromszögjel volt, ez azonban a hibamikrofonnál már a rendszer átvitele, és a különböző nemlinearitások miatt már csak hasonlított egy háromszögjelre. A kísérlet időtartománybeli alakulását mutatja a 8.6. ábra. A zajcsökkentés körülbelül a 11. s után lett bekapcsolva.



8.6. ábra. A Filtered-X LMS algoritmus működése az időtartományban háromszögjel esetén, mintavételi frekvencia 2 kHz, a zajscsökkentés 10 s után lett bekapcsolva

Az ábrán jól látható, hogy ebben az esetben a beállási idő javult, körülbelül 5 s. Ez annak köszönhető, hogy a lehető leggyorsabb beállást elérendő, minden kísérlet előtt módosítottam a bátorsági tényezőt. Ebben az esetben a bátorsági tényezőt nagyobbra véve a zajscsökkentő rendszer konvergenciasebessége nagyobb lett. A kísérlet eredményének frekvenciatartománybeli képét a 8.7. ábra mutatja.



8.7. ábra. A Filtered-X LMS algoritmus működése a frekvenciatartományban háromszögjel esetén, mintavételi frekvencia 2 kHz

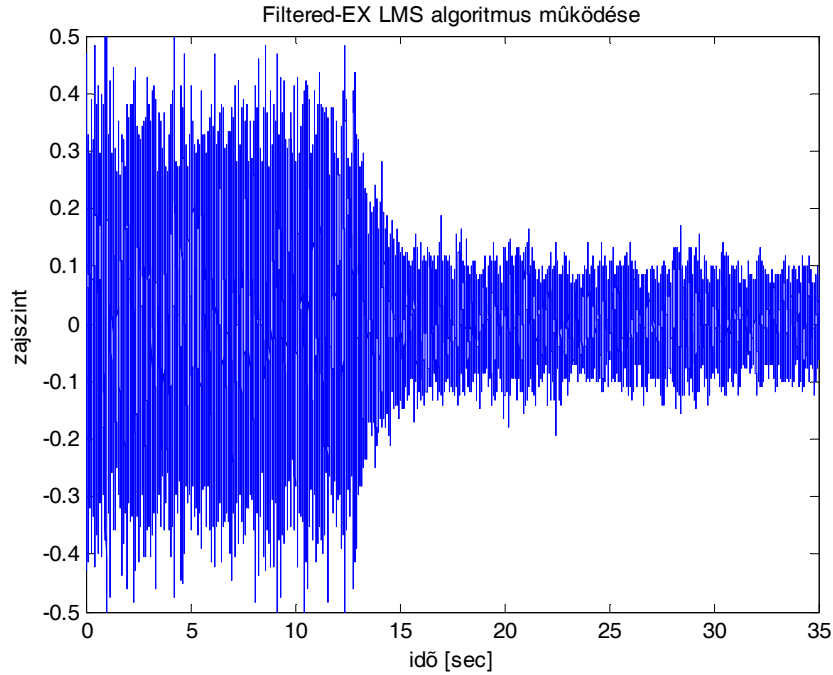
Az ábrán jól megfigyelhető a háromszögjelre jellemző spektrum. A jel előállításában elvileg csak a páratlan harmonikusok vesznek részt, a spektrumban megjelenő egyéb felharmonikusok a különböző nemlinearitásokat, esetleg a függvény generátor hibáját mutatják. Az ábráról leolvasható, hogy az alapharmonikusra nézve megközelítőleg 45 dB, a harmadik harmónikusra 60 dB, az ötödikre 30 dB az elnyomás.

Összegezve: a Filtered-X LMS algoritmus jól használható mind sztochasztikus, mind periodikus jelek elnyomására. Az eljárás hátránya kis konvergenciasebessége, ami már a szimulációkban is megmutatkozott.

### 8.3. *Filtered-EX LMS algoritmus*

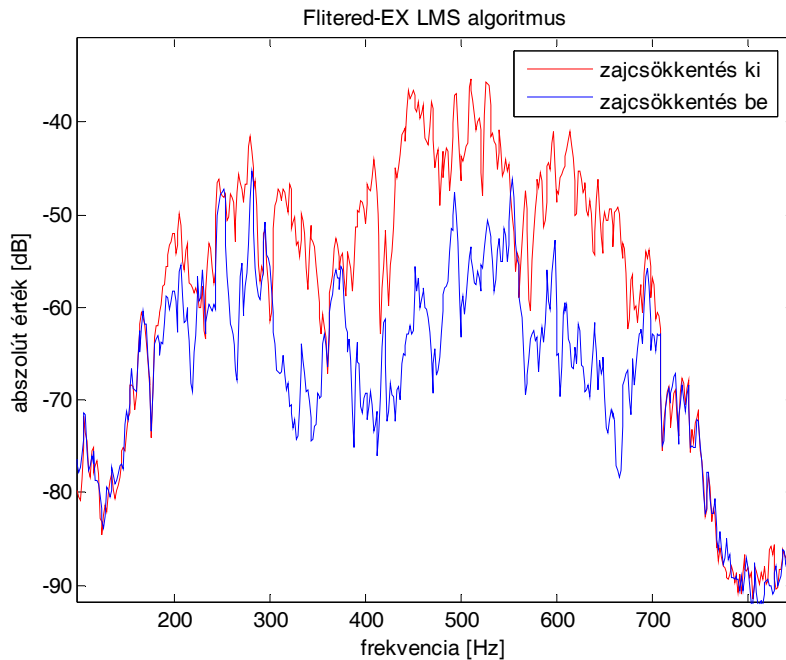
Az első mérésnél ennél az eljárásnál is azt vizsgáltam, hogyan használható az eljárás a mintavételi frekvenciához képest széles sávú zaj elnyomására. Az elnyomandó jel és a mérési elrendezés ugyanaz volt, mint amit a Filtered-X LMS algoritmusnál használtam. Az időtartománybeli eredmény látható a 8.8. ábrán. A zajcsökkentés a 13 s után lett bekapcsolva.





8.8. ábra. A Filtered-EX LMS algoritmus működése az időtartományban szélessávú zaj esetén, mintavételi frekvencia 2 kHz, a zajcsökkentés 13 s után lett bekapcsolva

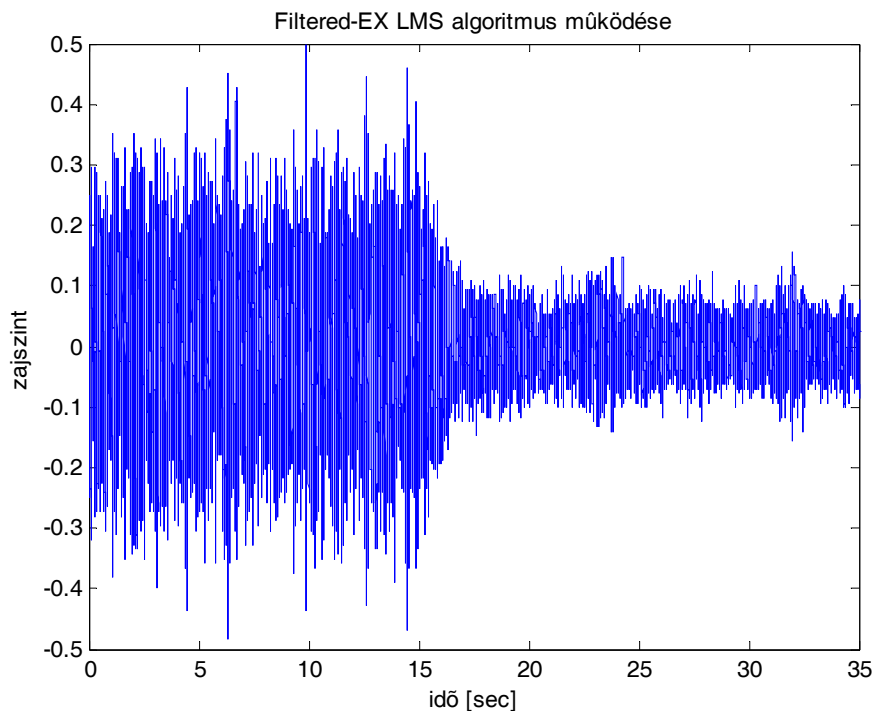
A kísérlet jól mutatja az Filtered-EX és a Filtered-X LMS algoritmus konvergenciasebessége közötti különbséget. Míg a Filtered-X LMS esetén a beállási idő ugyanilyen körülmények között 10 s volt, addig Filtered-EX LMS eljárást alkalmazva a konvergenciasebesség körülbelül háromszorosára nőtt, a beállítás csak 3 s-ot vett igénybe. A kísérlet eredményét a frekvenciatartományban is ábrázoltam, ahogyan azt a 8.9 ábra mutatja.



8.9. ábra. A Filtered-EX LMS algoritmus működése a frekvenciatartományban szélessávú zaj esetén, mintavételi frekvencia 2 kHz

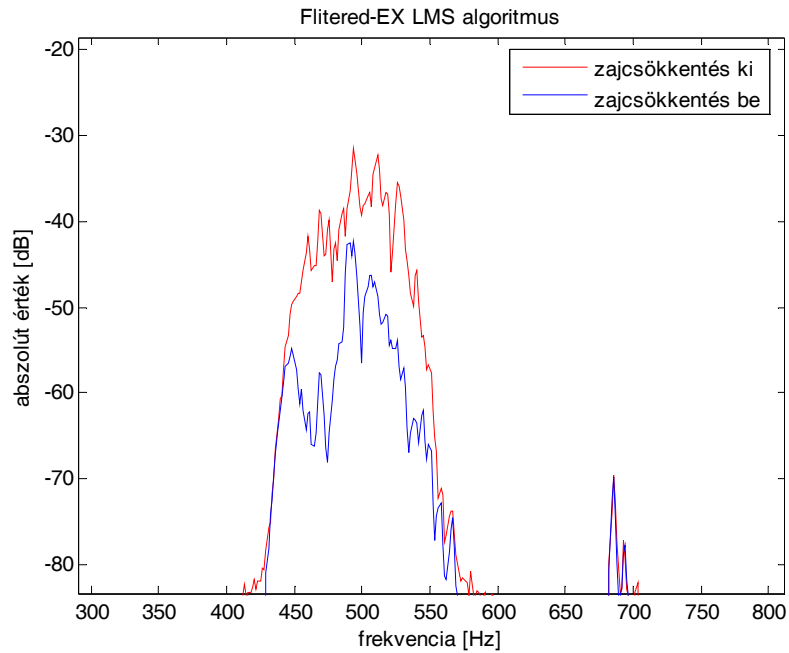
Az ábra jól mutatja, hogy széles tartományban igen jó az elnyomás, jobb, mint amit a Filtered-X LMS algoritmus esetén láttunk ugyanerre a kísérletre. Ezt az magyarázza, hogy azokon a frekvenciákon, ahol Filtered-X LMS esetén nem tapasztaltunk jelentős elnyomást, a konvergenciasebesség olyan kicsi lehetett, hogy még nem érte el a minimális elnyomási állapotot.

A második esetben az elnyomandó jel ugyanaz a 450 és 550 Hz közötti fehér spektrumú zaj volt, mint az előző alfejezetben a Filtered-X LMS algoritmusnál. Ennek a kísérletnek az időtartománybeli képe látható a 8.10. ábrán. A zajcsökkentés a 15. másodpercben lett bekapcsolva.



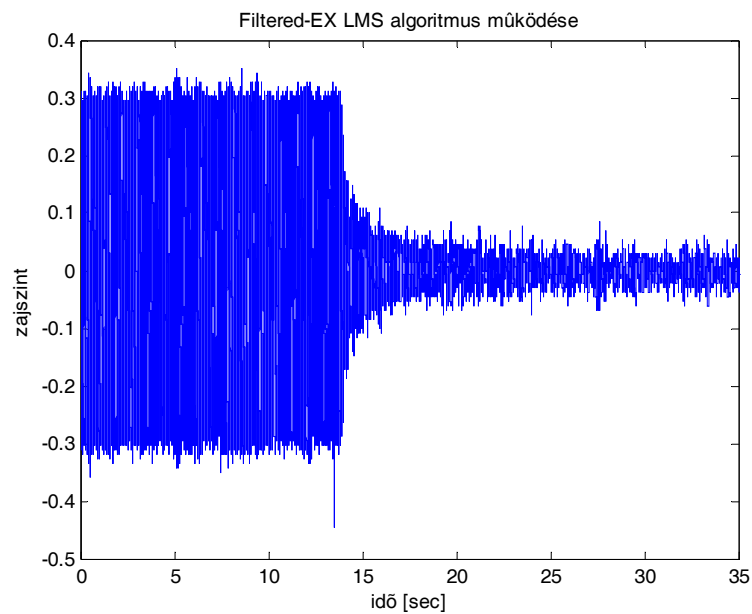
8.10. ábra. A Filtered-EX LMS algoritmus működése az időtartományban keskeny sávú zaj esetén, mintavételi frekvencia 2 kHz, a zajcsökkentés 15 s után lett bekapcsolva

A mérést ugyanaz a gyors, körülbelül 2 s-os beállítás jellemzi, mint az előzőt. Ugyanennek a mérésnek a frekvenciatartománybeli képe az állandósult állapotokra vonatkoztatva látható a 8.11. ábrán. A mérést ebben az esetben is ugyanaz a 680 Hz körüli frekvencián levő zaj zavarta, mint amit az előbbi alfejezetben a Filtered-X LMS algoritmusnál láthattunk. Az elnyomás ennél az eljárásnál is körülbelül egyenletes a frekvenciatartományban.



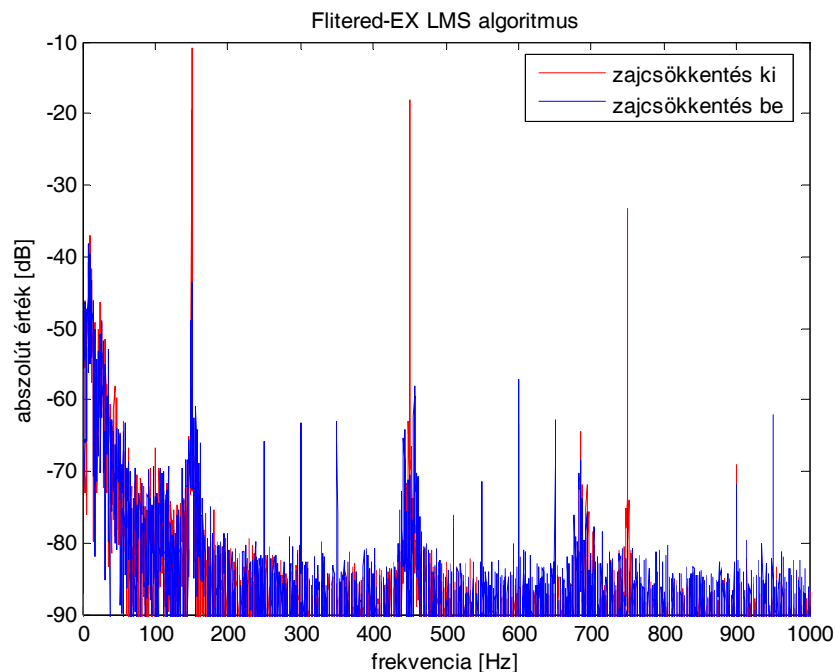
8.11. ábra. A Filtered-EX LMS algoritmus működése a frekvenciatartományban keskeny sávú zaj esetén, mintavételi frekvencia 2 kHz

A harmadik mérést periodikus jellel végeztem, az elnyomandó jel 150 Hz alapharmonikus frekvenciájú háromszögjel volt. Az időtartománybeli képet a 8.12. ábrán láthatjuk. A zajcsökkentés a 14. s-ban lett bekapcsolva.



8.12. ábra. A Filtered-EX LMS algoritmus működése az időtartományban háromszögjel esetén, mintavételi frekvencia 2 kHz, a zajcsökkentés 14 s után lett bekapcsolva

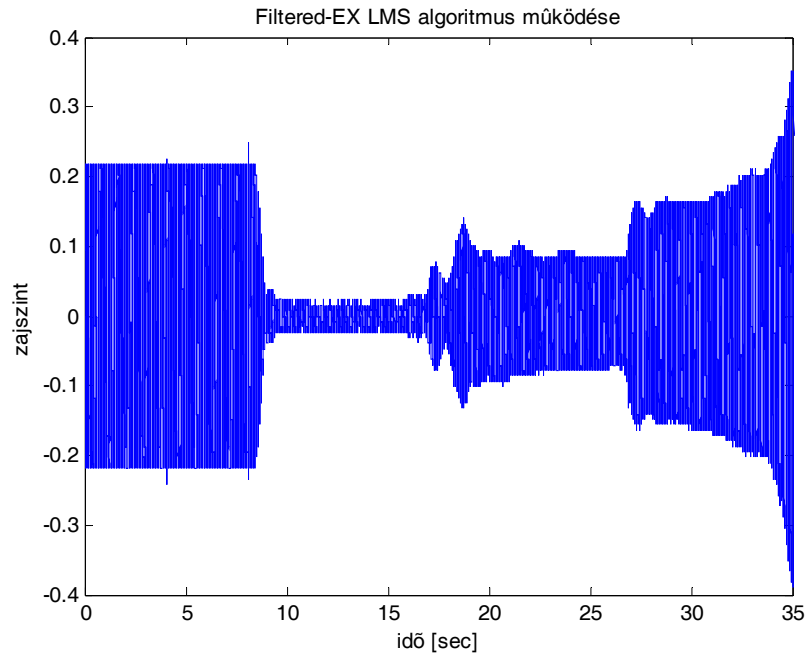
A beállási tulajdonságok nem változtak, gyors, stabil rendszer működését mutatja az ábra. Ugyanez a mérési eredmény a frekvenciatartományban a 8.13. ábrának megfelelő. Az ábra, akárcsak a 8.7. ábra igen jó elnyomást mutat periodikus jel esetén.



8.13. ábra. A Filtered-EX LMS algoritmus működése a frekvenciatartományban háromszögjel esetén, mintavételi frekvencia 2 kHz

A negyedik kísérletben azt mutatom meg, hogyan reagál az on-line identifikáció nélkül működő Filtered-EX LMS algoritmus arra, ha a másodlagos út átviteli függvénye megváltozik. A változást úgy valósítottam meg, hogy a hibamikrofont addigi pozíciójából körülbelül fél méterrel hátrébb mozdítottam, és derékszögben elforgattam. Ez az addig stabil rendszer működését a 8.14. ábrán látható módon befolyásolta.

Az elnyomandó jel ebben az esetben is az előbbi háromszögjel volt, a mérés kezdetekor a mikrofon még a korábbi pozíciójában volt, a zajcsökkentést a 8. s-ban kapcsoltam be, az átviteli függvényt pedig a 17. s-ban változtattam meg. Az ábrán jól követhető a rendszer reakciója a változásra. Az addig stabil zajcsökkentő struktúra a változás következtében instabillá válik, aminek eredményeképpen a hibamikrofon jele jóval az eredeti zajszint fölé nőtt.



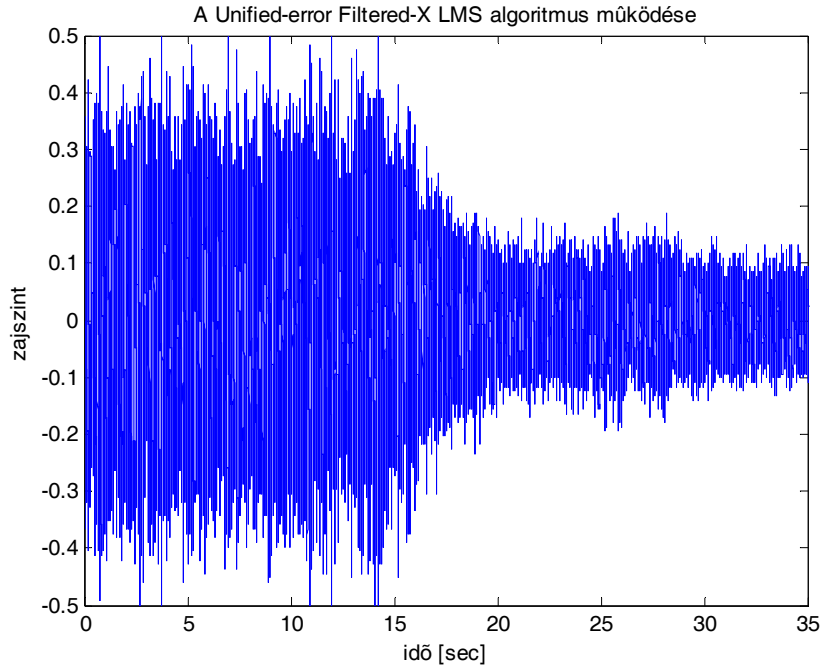
8.14. ábra. Filtered-EX LMS algoritmus működése változó másodlagos átviteli út esetén, mintavételi frekvencia 2 kHz, a zajcsökkentés a 8. s-ban lett bekapcsolva, a változás a 17. s-ban történt.

Összegezve a mérési eredményeket, az on-line identifikáció nélkül működő Filtered-EX LMS algoritmus a Filtered-X LMS-nél sokkal nagyobb konvergenciasebességű, a másodlagos átviteli út változását azonban on-line identifikáció nélkül csak kis változásokig tudja tolerálni.

#### 8.4. *Unified-error Filtered-X LMS algoritmus*

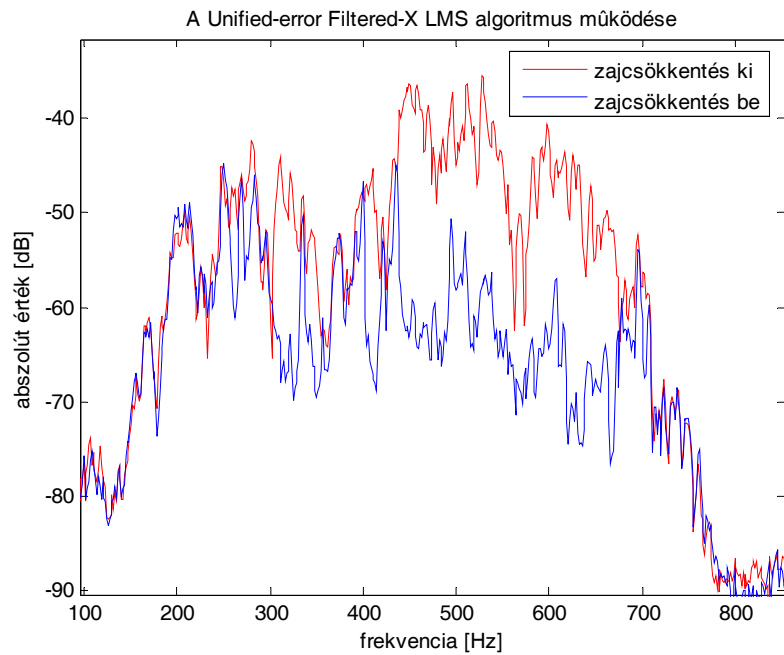
A Unified-error Filtered-X LMS algoritmus az általam az LMS családból választott on-line identifikációs algoritmus. A mérésekkel két dolgot kellett igazolnom. Az első, hogy képes zajcsökkentésre periodikus és sztochasztikus zajok esetén is. A második, hogy megmutassam, változó másodlagos átviteli út esetén is stabil marad a rendszer, és biztosítani tudja a zajcsökkentést. Ezeknek megfelelően ugyanazokat a méréseket végeztem el, mint az Filtered-EX LMS algoritmusra.

Az első mérést tehát szélessávú zajra végeztem. Az időtartománybeli mérési eredmény a 8.15. ábrán látható. A zajcsökkentés a 15. s után lett bekapcsolva.



8.15. ábra. A UFX LMS algoritmus működése az időtartományban szélessávú zaj esetén, mintavételi frekvencia 2 kHz, a zajcsökkentés 15 s után lett bekapcsolva

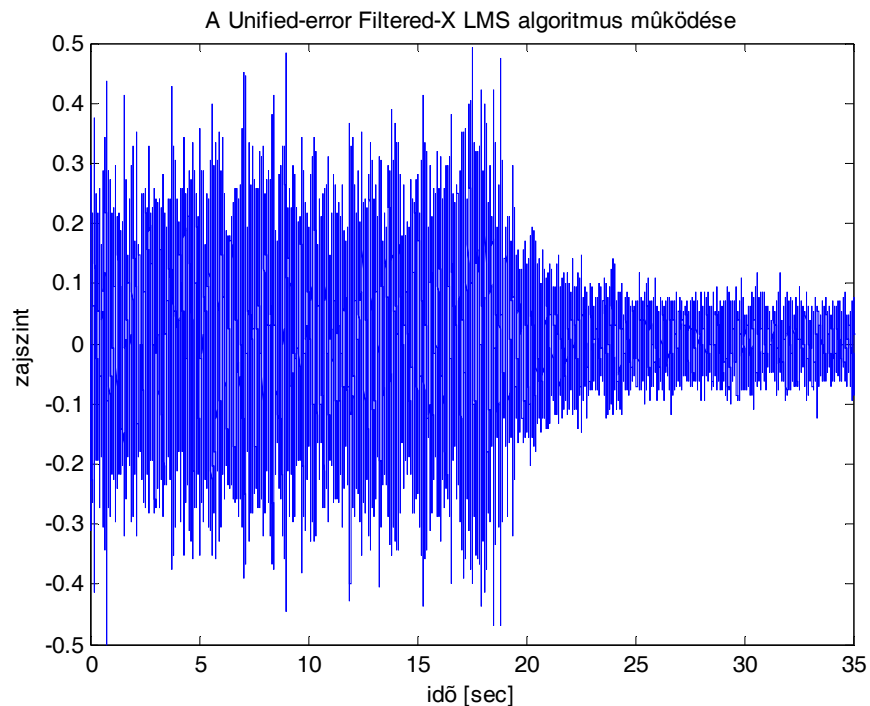
Az ábráról leolvasható, hogy az algoritmus beállási ideje körülbelül 5 s, ami 3s-al marad el az Filtered-EX LMS algoritmus esetén mért 2 s-tól. Ugyanennek a mérésnek az állandósult állapotbeli spektrális képe látható a 8.16. ábrán.



8.16. ábra. A UFX LMS algoritmus működése a frekvenciatartományban szélessávú zaj esetén, mintavételi frekvencia 2 kHz

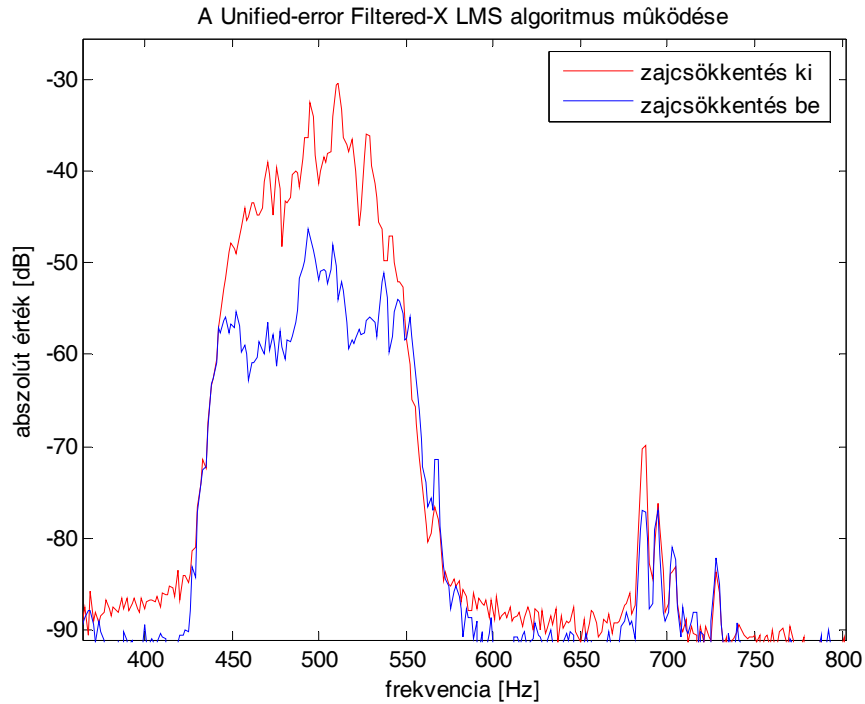
Az ábra jól mutatja, hogy a frekvenciatartománybeli eredmény nagyon hasonló az Filtered-EX LMS esetén tapasztalhoz, széles sávban hatékony zajcsökkentés tapasztalható.

A második mérésben az elnyomandó jel keskeny sávú fehér zaj volt. Az erre kapott időtartománybeli eredmény látható a 8.17. ábrán. A zajcsökkentés a 18. s után lett bekapcsolva.



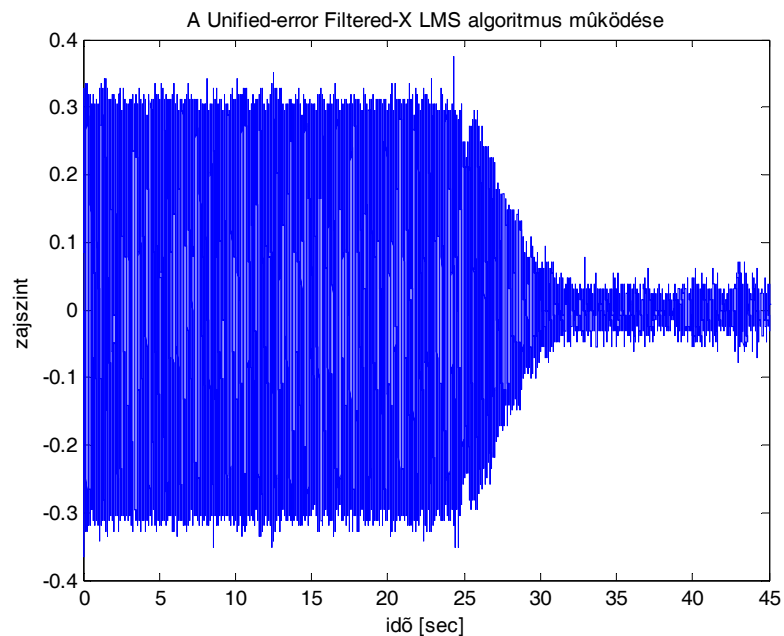
8.17. ábra. A UFX LMS algoritmus működése az időtartományban keskeny sávú zaj esetén, mintavételi frekvencia 2 kHz, a zajcsökkentés 18 s után lett bekapcsolva

A beállási idő nem változott, továbbra is 5 s körüli, az előző alfejezetekben bemutatott két algoritmus között van. Ugyanennek a mérésnek a spektrális képe látható a 8.18. ábrán. A zajcsökkentés a teljes sávban jelentős mértékű, de a mérés során továbbra is jelen volt a 680 Hz körüli zavarhatás, ami a mostani mérés során valószínűleg beszűrődött a referenciajelbe, azért csökkent a zajcsökkentés bekapcsolásakor.



8.18. ábra. A UFX LMS algoritmus működése a frekvenciatartományban keskeny sávú zaj esetén, mintavételi frekvencia 2 kHz

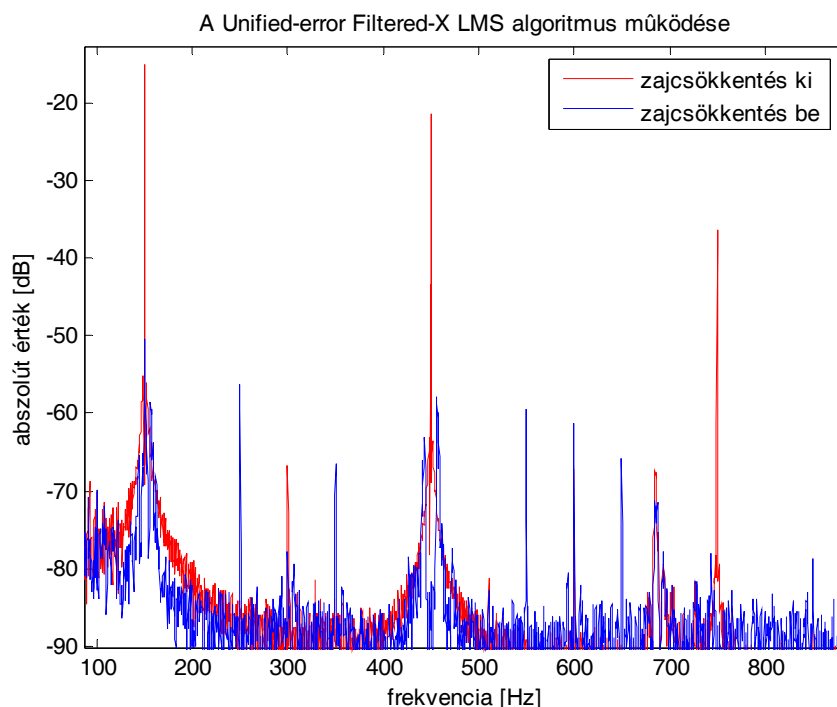
A harmadik mérést ebben az esetben is periodikus jelre végeztem, az elnyomandó jel az előző alfejezetekben bemutatott háromszögjel volt. A mérés alakulását az idő függvényében a 8.19. ábra mutatja. A zajcsökkentést a 26. s után kapcsoltam be.



8.19. ábra. A UFX LMS algoritmus működése az időtartományban háromszögjel esetén, mintavételi frekvencia 2 kHz, a zajcsökkentés 26 s után lett bekapcsolva



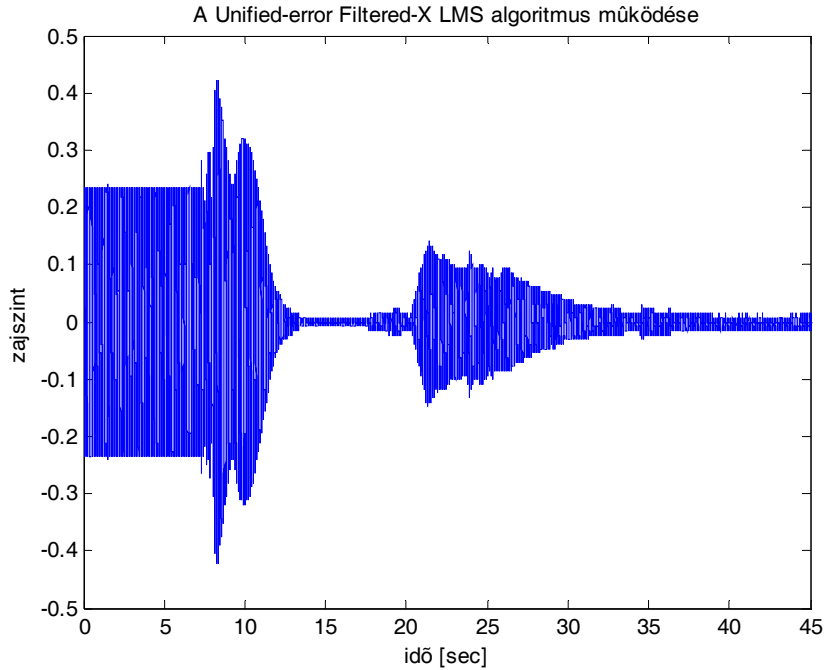
Az elnyomás ebben az esetben is jelentős, a beállítás gyorsabb, 3-4 s körüli, ami a megnövelt bátorsági tényezőnek köszönhető. Az állandósult állapotbeli spektrumok közötti különbséget a 8.20. ábra mutatja.



8.20. ábra. A UFX LMS algoritmus működése a frekvenciatartományban háromszögjel esetén, mintavételi frekvencia 2 kHz

Az ábra nagyon hasonlít a 8.13. ábrán látható spektrumra, az elnyomás tehát hasonlóan nagy, mint az Filtered-EX LMS algoritmus esetén.

A negyedik kísérletben megvizsgáltam, hogyan reagál az algoritmus a másodlagos átviteli út változásaira. A változást ebben az esetben is úgy valósítottam meg, hogy a hibamikrofont addigi pozíciójából egy fél méterrel hátrébb mozdítottam, és derékszögben elforgattam. A zajcsökkentést a 8. s-ban kapcsoltam be. A rendszer ki lengések után beállt az állandósult állapotba. Ezt követően a 18. s-ban megváltoztattam az átviteli függvényt, ennek hatására megnőtt a zajszint a hibamikrofonnál, de a rendszer végül, a 35. s körül megtalálta a stabil állapotot, de a zajszint ez idő alatt sem ment az eredeti fölé, sőt jóval alatta maradt, tehát az algoritmus változó körülmények között is biztosítani tudta az elnyomást.



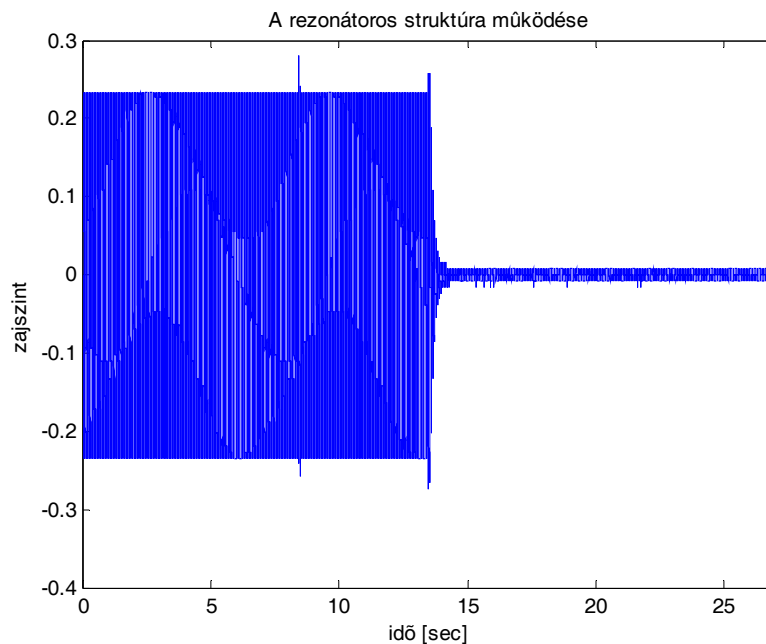
8.21. ábra. UFX LMS algoritmus működése változó másodlagos átviteli út esetén, mintavételi frekvencia 2 kHz, a zajcsökkentés a 8. s-ban lett bekapcsolva, a változás a 18. s-ban történt.

### 8.5. Rezonátoros zajcsökkentő struktúra

A rezonátoros struktúra periodikus jelek elnyomására lett kifejlesztve. A többi algoritmussal való összehasonlíthatóság kedvéért tehát a méréseket az előző fejezetben bemutatott háromszögjellel mint elnyomandó jellel végeztem. A referenciajelet rezonátoros esetben elektronikus úton vettem, aminek azért van létjogosultsága, mivel a rezonátoros struktúrának csak az elnyomandó jel egyik harmonikusára van szüksége ahhoz, hogy a rezonátorpozíciókat megfelelően hangolja. Ez pedig a gyakorlatban általában elérhető valamilyen mérési módszerrel, például forgó motornál egy szenzor segítségével rögzíteni a fordulatszámot, és ebből a fordulatszámmal arányos elnyomandó jel alaphfrekvenciája számolható.

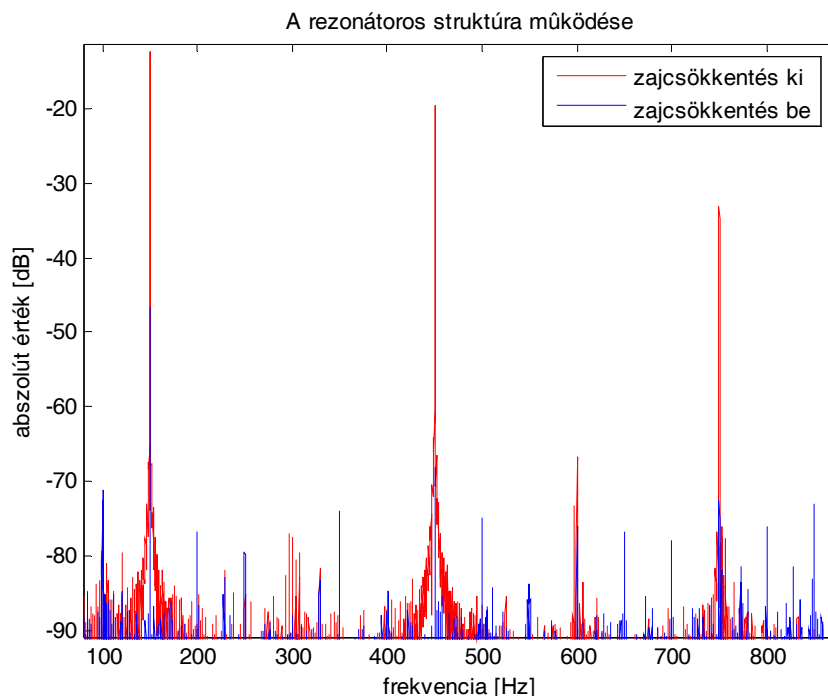
A rezonátoros struktúra működésének ellenőrzésére több mérést végeztem, az első esetben az elnyomandó jel a fentiekben említett 150 Hz alaphfrekvenciájú háromszögjel volt. A második mérésben megvizsgáltam, hogyan reagál a rezonátoros zajcsökkentő struktúra a másodlagos átviteli út megváltozására (a változást az előző alfejezetekben ismertetett módon idéztem elő). A harmadik mérést sweepelő szinuszos elnyomandó jelre végeztem, hogy megmutassam, miképp követi a rezonátoros zajcsökkentő struktúra az elnyomandó jel frekvenciájának változását.

Az első mérés időtartománybeli eredménye látható a 8.22. ábrán. A zajcsökkentés a 13. s-ban lett bekapcsolva.



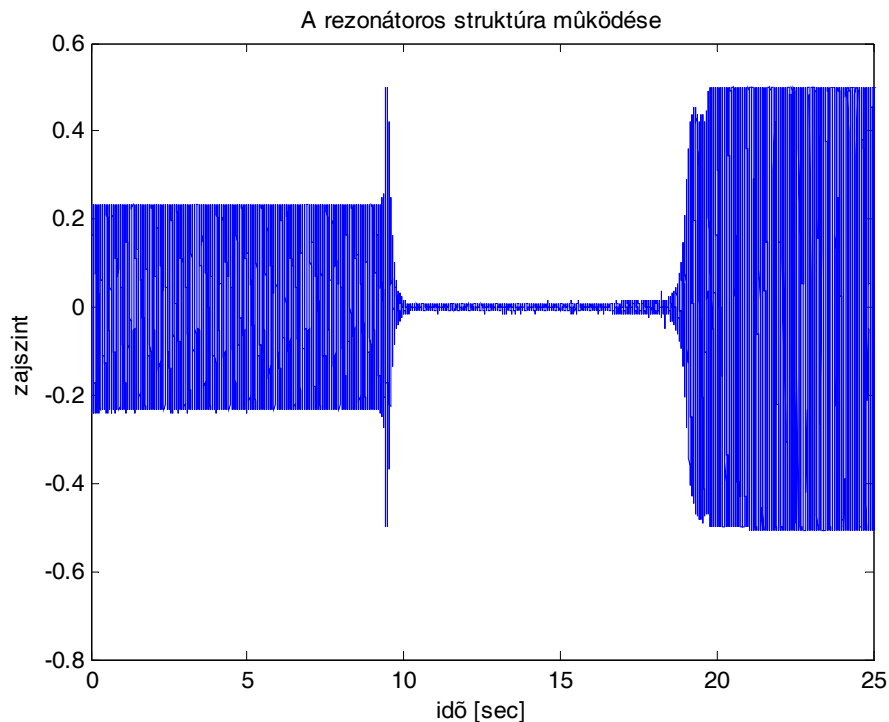
8.22. ábra. A rezonátoros struktúra működése az időtartományban háromszögjel esetén, mintavételi frekvencia 2 kHz, a zajcsökkentés 13 s után lett bekapcsolva

Az ábra jól mutatja, hogy a rezonátoros struktúrával érhető el a leggyorsabb beállítás. Az állandósult állapotot a rendszer már 1 s-mal a zajcsökkentés bekapcsolása után eléri. A mérés frekvenciatartománybeli képét mutatja a 8.23. ábra. Az elnyomás mértéke egyezik az előző alfejezetekben tapasztaltakkal.



8.23. ábra. A rezonátoros struktúra működése a frekvenciatartományban háromszögjel esetén, mintavételi frekvencia 2 kHz

A második méréssel azt vizsgátam, hogyan reagála rendszer a másodlagos út átviteli függvényének megváltozására. Ezt mutatja a 8.24. ábra.

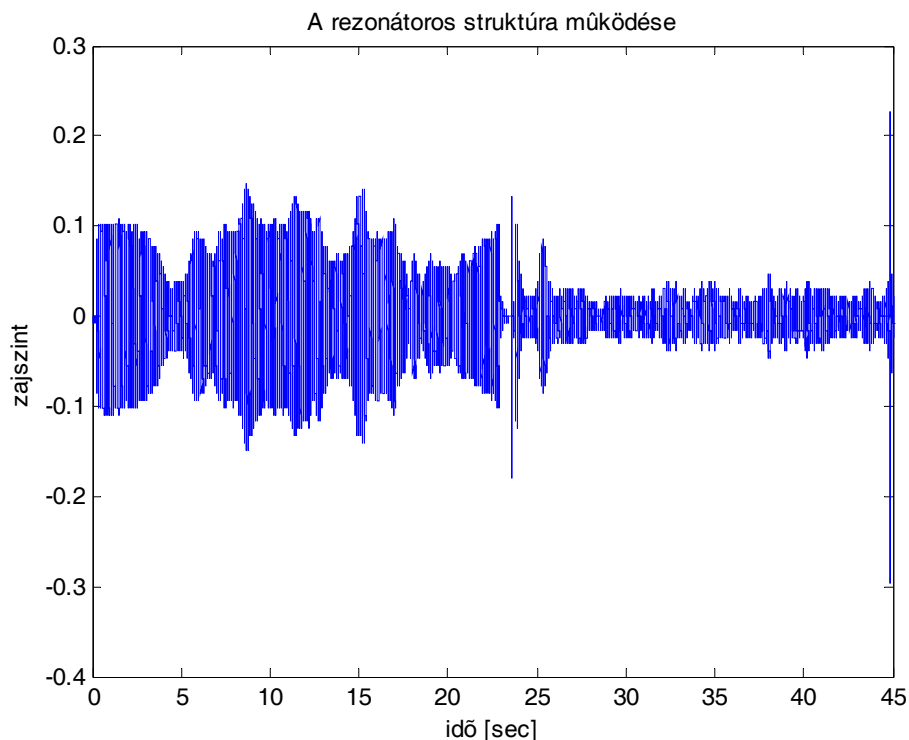


8.24. ábra. A rezonátoros struktúra működése változó másodlagos átviteli út esetén, mintavételi frekvencia 2 kHz, a zajcsökkentés a 8. s-ban lett bekapcsolva, a változás a 18. s-ban történt.

Mivel ebben az esetben sincs on-line identifikáció, a rendszer a 18. s-ban bekövetkező változás hatására instabillá válik. A beavatkozó jel exponenciálisan nőni kezd.

A negyedik kísérletben a rezonátoros struktúra frekvenciakövető tulajdonságát vizsgáltam. A sweepelő szinuszos jel 100 és 200 Hz között sweepelt lineárisan, 20 s alatt. A mérés első felében a zajcsökkentés ki volt kapcsolva, majd a 23. s-tól bekapcsoltam, és így is felvettem az eredményt. A mérés első fele tehát zajcsökkentés nélkül, míg második fele zajcsökkentéssel lett felvéve. Az eredményt a 8.25. ábra mutatja. Az ábra jól mutatja, hogy a rezonátoros struktúra nagyon jól követi az elnyomandó jel frekvenciájának változásait, ezért hatékonyan használható olyan alkalmazásokban, ahol a zaj frekvenciája változik, mint például motorzaj esetén a gyorsításkor bekövetkező frekvenciaváltozás.

Összegezve: a rezonátoros struktúra hatékonyan használható periodikus jelek elnyomására. Elnyomási tulajdonságai megegyeznek az LMS családdal, de konvergenciasebessége nagyobb, ezért az elnyomandó jel változásait hatékonyabban tudja követni, mint az LMS alapú eljárások.

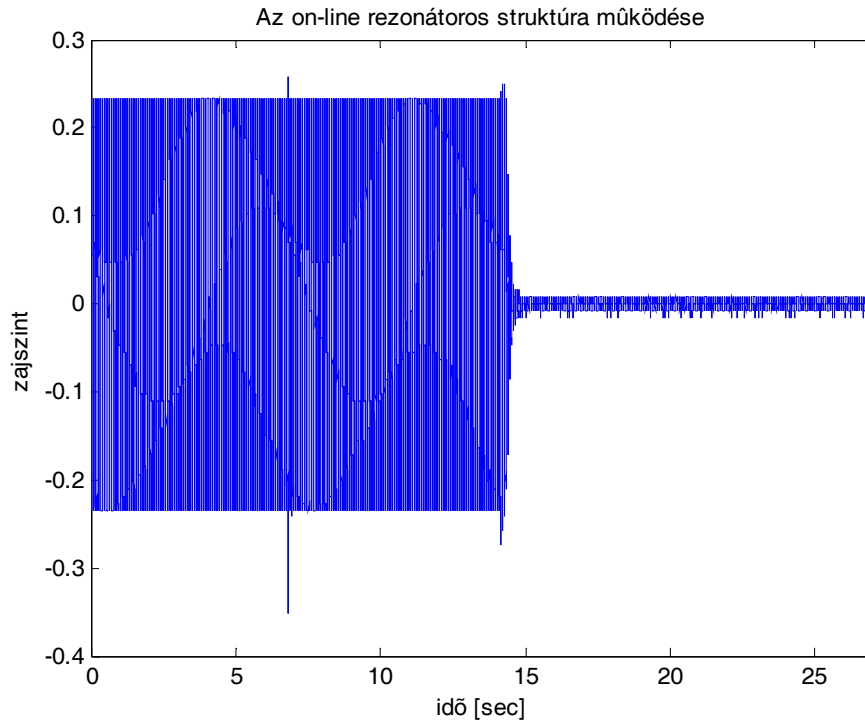


8.25. ábra. A rezonátoros struktúra működése változó sweepelő szinuszos elnyomandó jel esetén, mintavételi frekvencia 2 kHz, a zajcsökkentés a 23. s-ban lett bekapcsolva.

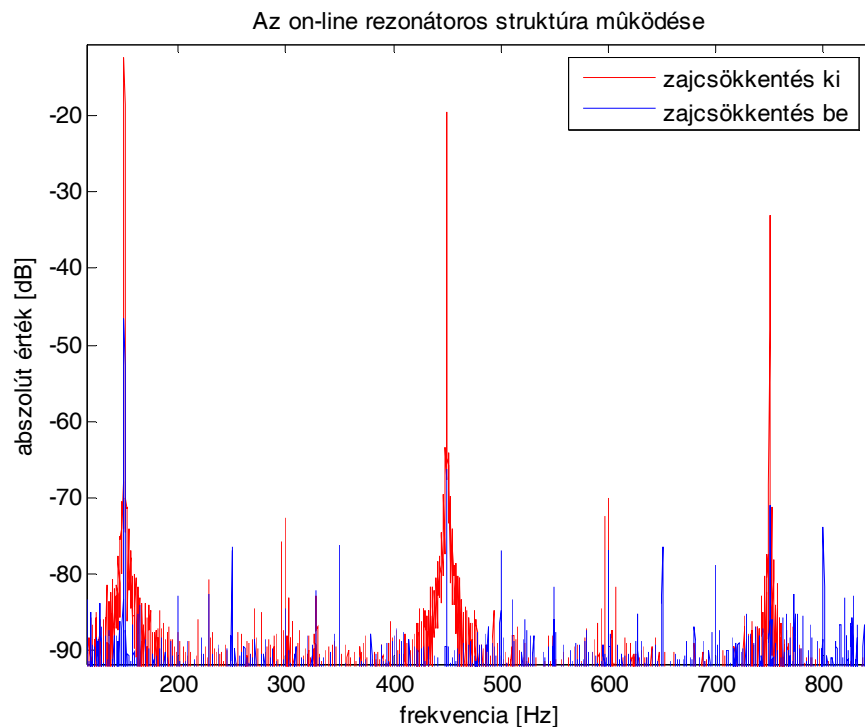
## 8.6. *Rezonátoros zajcsökkentő struktúra on-line identifikációval*

Az on-line identifikációs rezonátoros struktúra mérésekor a cél az volt, hogy megmutassam, konvergenciasebessége és elnyomási tulajdonságai egyeznek az on-line identifikáció nélkül működő rezonátoros struktúráéval, de azzal ellentétben jól tudja tolerálni a másodlagos átviteli út változásait.

Az első mérésben az elnyomandó jel a korábbiakban alkalmazott háromszögjel volt. A mérés időbeli lefutása a 8.26. ábrán követhető végig. A zajcsökkentést a 14. s után kapcsoltam be, az eredmény pedig 1 s-os beállási idő, ugyanúgy, mint az előző alfejezetben. A frekvenciatartománybeli képet a 8.27. ábra mutatja. Az elnyomási tulajdonságok szintén egyeznek a rezonátoros struktúránál tapasztaltakkal.

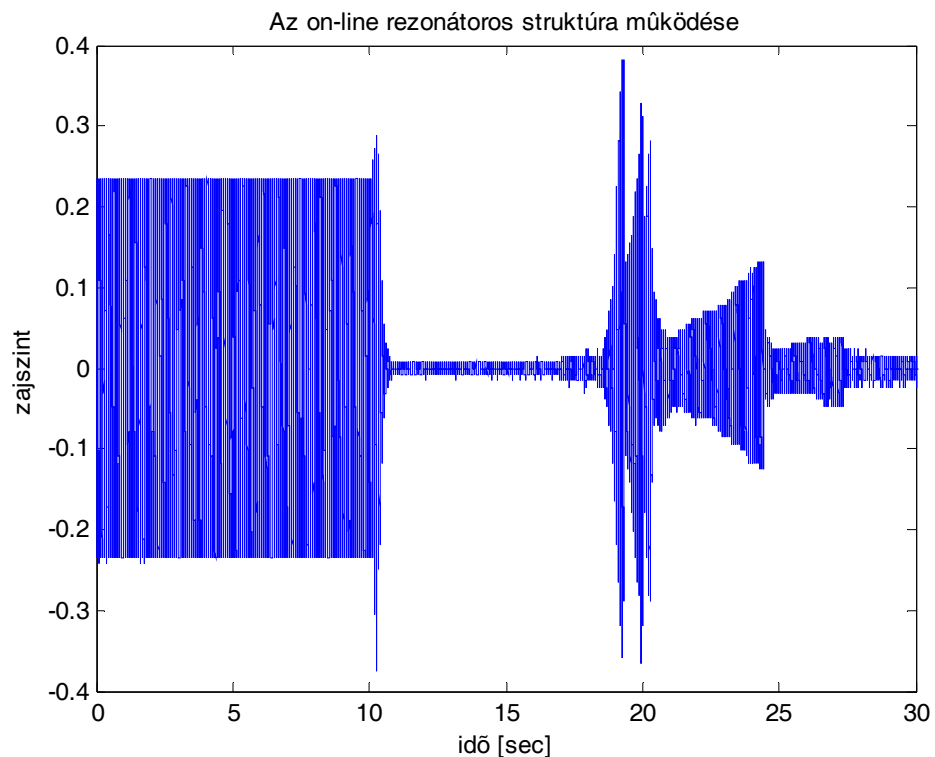


8.26. ábra. Az on-line rezonátoros struktúra működése az időtartományban háromszögjel esetén, mintavételi frekvencia 2 kHz, a zajcsökkentés 14 s után lett bekapcsolva



8.27. ábra. Az on-line rezonátoros struktúra működése a frekvenciatartományban háromszögjel esetén, mintavételi frekvencia 2 kHz

A harmadik mérésben változó átviteli tulajdonságok esetén vizsgáltam a zajcsökkentő struktúra működését. Az elnyomandó jel és a változtatás megegyezett az előző alfejezetekben az ilyen a típusú mérésre leírtakkal. A mérés időbeli lefutását mutatja a 8.28. ábra. A zajcsökkentés a 10. s-ban lett bekapcsolva, a változás a 17. s-ban történt. Az ábrán jól látható, hogy az on-line struktúra több instabil állapot után iterálva megtalálja a stabil állapotot. A UFX LMS algoritmussal összehasonlítva azt mondhatom, hogy stabilitás és konvergenciasebesség szempontjából a két algoritmus egyenértékű, azonban a UFX LMS algoritmusnál a hibamikrofon jele az átviteli függvény változásakor sem megy az eredeti zajszint fölé, míg az általam kifejlesztett struktúráról ugyanez nem mondható el.



8.28. ábra. Az on-line rezonátoros struktúra működése változó másodlagos átviteli út esetén, mintavételi frekvencia 2 kHz, a zajcsökkentés a 10. s-ban lett bekapcsolva, a változás a 17. s-ban történt.





## 9. Összefoglalás, kitekintés

Dolgozatom első felében áttekintettem az aktív zajcsökkentés irodalmát bemutatva a leggyakrabban használt, legismertebb zajcsökkentő eljárásokat. Az LMS algoritmus alapú zajcsökkentő eljárások ismertetésével kezdtem a szakirodalom tárgyalását. Ezek közül a legalapvetőbb zajcsökkentő eljárás a Filtered-X LMS algoritmus, amelynek az alapja tulajdonképpen egy LMS algoritmus, a másodlagos akusztikus átviteli út modelljével kiegészülve. Ebből az eljárásból fejlesztették ki a Filtered- $\epsilon$  LMS és a Filtered-EX LMS algoritmusokat, amelyek ugyanazt a problémát, azaz a Filtered-X LMS algoritmus konvergenciasebességének javítását tűzték ki célul. Az on-line identifikációs megoldások közül áttekintettem a fenti algoritmusok on-line identifikációval való kiegészítésének lehetőségét, majd bemutattam egy ezeknél hatékonyabb működésű, kevésbé ismert on-line identifikációval dolgozó aktív zajcsökkentő eljárást, a Unified-error Filtered-X LMS algoritmust.

A következő fejezetekben bemutattam egy periodikus zajok elnyomására a Méréstechnika és Információs Rendszerek Tanszéken kifejlesztett zajcsökkentő struktúrát, melynek alapja periodikus jelek koncepcionális modelljéhez illeszkedő megfigyelő tervezése. A továbbiakban ennek a struktúrának az on-line identifikációval való kiegészítésének lehetőségét vizsgáltam, majd bemutattam egy alapötletből kiindulva egy általam kifejlesztett algoritmust, melynek működési lényege a következő. Az algoritmus detektálja, ha a rendszer instabil vagy a stabilitás határhelyzetében van. Ekkor beavatkozik, és iteratív úton  $\pi/2$  pontossággal meghatározza az akusztikus átvitel fázisát, így biztosítva a rendszer stabilitását.

Dolgozatom utolsó harmadában bemutattam a megvalósított zajcsökkentő eljárásokat, majd korábban bemutatott elméleti és szimulációs eredményeimet mérési eredményekkel támasztottam alá. Az eredmények: periodikus jelek alapharmonikusára nézve 45 dB csillapítást, míg zajra 15 dB csillapítást értem el.

A jövőbeni továbbfejlesztéseknek több iránya is lehetséges. Egyik irányvonal lehet a rezonátoros struktúrához egycsatornás esetre kifejlesztett on-line identifikációs eljárás többcsatornás esetre való kiterjesztési lehetőségének vizsgálata. Másik irányként kínálkozik a Unified-error Filtered-X LMS algoritmus rezonátoros struktúrára való átültetési lehetőségének vizsgálata, valamint ennek az algoritmusnak a többcsatornás esetre való továbbfejlesztése.



## 10. Felhasznált irodalom

- [1] Widrow, Walach: „Adaptive Inverse Control”, Prentice Hall PTR, Upper Saddle River, New Jersey 07458, 1996
- [2] BME MIT Tanszéki Munkaközösség: „Digitális jelfeldolgozás”, Budapesti Műszaki Egyetem, segédlet, 2004
- [3] Nozomu Saito, Toshio Sone, Tomohiko Ise, Masaichi Akiho: „Conditions for optimal on-line identification of secondary path in active noise control systems”, *Active '95*, Newport Beach, CA, USA, 1995
- [4] Hyoun-Suk Kim, Youngjin Park: „Unified-error filtered-X LMS algorithm for on-line active control of noise in time-varying environment”, *Active '97*, Budapest, Hungary, 1997
- [5] L. J. Eriksson, T. A. Laak, M. C. Allie: „On-line secondary path modelling for FIR and IIR adaptive control in the presence of acoustic feedback”, *Active '99*, Newport Beach, CA, USA, 1999
- [6] Sujbert László: „A new filtered LMS algorithm for active noise control” *Active '99*, Newport Beach, CA, USA, 1999
- [7] Elek Kálmán: „Adaptív jelfeldolgozás”, Budapesti Műszaki Egyetem, jegyzet, 2005
- [8] S. J. Elliott, P.A. Nelson: „Active noise control”, *IEEE Signal Processing Magazine*, Vol. 10, No. 4. pp. 12-35. October, 1993
- [9] S. M. Kuo, D. R. Morgan: „Active Noise Control: A Tutorial Review”, *Proceedings of the IEEE*, vol. 87. No. 6. pp 943-973, June, 1999
- [10] G. Péceli: „A common structure for recursive discrete transforms”, *IEEE Trans. Circuits Syst.* Vol. CAS-33. pp. 1035-1036, October, 1986
- [11] L. Sujbert, G. Péceli, Gy. Simon: „Resonator based non-parametric identification of linear systems”, in *IEEE Trans. On Instrumentation and Measurement*, vol. 54. No. 1. pp. 386-390. February 2005
- [12] L. Sujbert, G. Péceli: „Periodic noise cancellation using resonator based controller”, in *1997 Int. Symp. On Active Control of Sound and Vibration, ACTIVE '97*, pp. 905-916. Budapest, Hungary, August 1997
- [13] <http://www.tinyos.net>
- [14] <http://www.tinyos.net/nest/doc/tutorial/>

- [15] [http://www.xbow.com/Support/Support\\_pdf\\_files/XBOW\\_Smart\\_Dust\\_ProductInfoGuide.pdf](http://www.xbow.com/Support/Support_pdf_files/XBOW_Smart_Dust_ProductInfoGuide.pdf)
- [16] [http://www.analog.com/UploadedFiles/Application\\_Notes/399447663EE191.pdf](http://www.analog.com/UploadedFiles/Application_Notes/399447663EE191.pdf)
- [17] <http://pdfserv.maxim-ic.com/en/ds/MAX220-MAX249.pdf>
- [18] [http://www.semiconductors.philips.com/acrobat\\_download/datasheets/74HC\\_HCT74\\_3.pdf](http://www.semiconductors.philips.com/acrobat_download/datasheets/74HC_HCT74_3.pdf)
- [19] „ADSP-2106x SHARC User’s Manual (second edition)”, Analog devices Inc. P.O Box 9106 Norwood, 1997
- [20] „ADSP-2106x SHARC Reference Manual”, Analog devices Inc. P.O Box 9106 Norwood, 1997