

3. mérés

Adaptív szűrők vizsgálata

Sujbert László, Balogh Tibor
BME Méréstechnika és Információs Rendszerek Tanszék

1. A mérés célja

A mérés célja a korábbi szakirányú tanulmányok során megszerzett, a digitális jelfeldolgozás témakörébe tartozó adaptív szűrőkkel kapcsolatos ismeretek elmélyítése, azok gyakorlati vonatkozásainak megismerése. A mérés tematikája szerint a már meglévő elméleti ismeretek szükséges szintű elmélyítését azok gyakorlati alkalmazásainak vizsgálata követi.

A manapság jellemző tendenciák szerint egyre jobban terjed az adaptív jelfeldolgozás gyakorlati alkalmazása, így ez a témakör a villamosmérnök eszköztárát már nemcsak elméleti síkon, hanem gyakorlati szinten is bővíti. Napjainkban a gyártók termékpalettáján kínált jelfeldolgozó processzorok számítási teljesítménye egyre növekszik, ezzel párhuzamosan áruk egyre alacsonyabb, így elterjedtségük is növekvő tendenciát mutat, valamint olyan területeken is teret hódítanak, ahol eddig magas árú vagy nem elégséges komplexitásuk miatt nem alkalmazták őket. A mai, korszerű jelfeldolgozó processzorok tehát olyan gyakorlati problémák megoldására kínálnak lehetőséget, amelyek korábban elérhetetlenek és elképzelhetetlenek voltak – többek között – a bonyolult valós idejű jelfeldolgozás igénye miatt.

Mára tehát egyre több alkalmazásban van szükség, illetve lehetőség adaptív jelfeldolgozás beépítésére. Ezek legnagyobb csoportja az akusztikával vagy audio jelek átvitelével van szoros kapcsolatban. Az akusztikai tér átvitele ugyanis két fizikai hely – például hangszórók és mikrofonok, vagy természetes zajforrások és az emberi fül – között jellemzően változó. Az átviteli csatorna átvitele is jelentős mértékben változhat a sokszor fizikailag is egymástól távol lévő berendezések paramétereinek megváltozása miatt.

Számos gyakorlati feladatban alkalmazott algoritmus megfontolásai szerint szükség van viszont adott átvitel vagy átvitelek megfelelő pontosságú ismeretére. Ez az előző szempontokból kiindulva, az átvitel sztochasztikusan változó jellegű voltának ismeretében nem egyértelmű feladat. Lehetővé kell tenni tehát valamilyen módon egy ismeretlen és az időben változó átvitelű rendszer modelljének kielégítő pontosságú előállítását, valamint azt, hogy a modell a fizikai rendszer változásait valós időben megfelelő gyorsasággal kövesse.

A feladat tehát a modellillesztés, melynek része a modell parametrikus vagy strukturális meghatározása, vagy a rendszerrel hasonló viselkedésű, de eltérő felépítésű modell kialakítása. A mérés során ezeket a célokat egyrészt MATLAB szimulációk, másrészt egy áramkör és jelfeldolgozó kártya segítségével végzendő mérések szolgálják.

2. A mérés elméleti alapjai

A mérés elvégzéséhez mindenképpen szükséges az adaptív jelfeldolgozás elméleti hátterének átfogó ismerete, ugyanis a mérési feladatok egy részében jelfeldolgozó algoritmusok MATLAB-ban történő implementációját kell elvégezni, a többi feladatban pedig fizikailag realizált struktúrákon kell méréseket és paraméterhangolási feladatokat végezni.

2.1. A modellillesztés elméleti alapjai

A modellillesztés feladata – egy mondatban összefoglalva – egy ismeretlen rendszernek megfelelő modell strukturális meghatározása, illetve adott esetben az előzőleg ismertnek feltételezett struktúra paramétereinek meghatározása (*paraméterbecslés*). A gyakorlatban leginkább a paraméterbecslési feladattal találkozunk, melynek során az ismeretlen rendszerre legjobban illeszkedő paraméterekkel rendelkező, előzetesen adott struktúrájú modellt keressük.

A paraméterbecslés során a valóságos rendszer és a hozzá illesztett modell kimeneteinek eltérését adott hibakritérium alapján kívánjuk minimalizálni. A modellillesztés feladatköre ezen belül is két csoportra bontható. Az *identifikáció* célja az időben állandónak feltételezett rendszer paramétereinek meghatározása, míg az *adaptáció* feladata az időben változó rendszerparaméterek követése. Az identifikáció során tehát nagy pontosságú mérés elvégzésére van elvi lehetőség, ugyanis a rendszert a szükséges hosszú ideig megfigyelhetjük, a szükséges

mennyiségű adatot gyűjthetjük be róla. Az identifikáció ebből következően jellemzően időigényes folyamat. Az adaptáció során kevésbé fontos a nagyfokú pontosság és a modellnek a valóságos rendszerhez való pontos illeszkedése. Sokkal inkább lényeges, hogy a modell paramétereinek változtatásával a fizikai rendszer változásait követni tudjuk, és a modell a rendszerrel valós időben tudjon együtt mozogni. Természetesen a fizikai rendszer struktúrája nem minden esetben ismert, és paraméterei sem mindig mérhetőek, kizárólag a fizikai rendszer és a modell kimenetének eltérése adhat támpontot a modell paramétereinek korrigálásához. Ehhez az esetek döntő többségében le kell mondanunk a nagy pontosságú modellről, és helyette egy egyszerűbb, a gyakorlatban könnyebben kezelhető struktúrát kell alkalmaznunk, mely még elfogadható a valóságos rendszer modellezésére.

A modellillesztés egy speciális esete a regressziószámítás feladata, amelynek célja bizonyos változók között fennálló determinisztikus kapcsolat meghatározása. Ezen változók a gyakorlatban legtöbbször bizonyos rendszerek be- és kimenetei. A valóságos rendszerek, melyekre modellt kívánunk illeszteni, a bemenetük és kimenetük között azonban nem tisztán determinisztikus függvénykapcsolatot valósítanak meg, mert a kimenet rendszerint zajjal terhelt, tehát a determinisztikus mellett sztochasztikus komponenst is tartalmaz. Az illesztendő modell ellenben kizárólag determinisztikus függvénykapcsolatot realizál, melynek kialakítása tipikusan bizonyos szabad paraméterek valamilyen szempontból megválasztott elv alapján történő beállítását jelenti a mérések után rendelkezésre álló bemenet-kimenet adatpár-sorozat alapján.

A paraméterek hangolásához egy a valóságos rendszer és az illesztett modell kimeneteinek különbségétől függő ún. *költségfüggvényt* definiálunk, és a paraméterek hangolásával ennek minimális értékét kísérreljük meg beállítani. Célunk tehát annak elérése, hogy a modell kimenete a lehető legjobban közelítse a valóságos rendszer kimenetét, a „lehető legjobban” kritérium egzakt leírása pedig a költségfüggvény vagy hibakritérium-függvény kifejezése:

$$\varepsilon = E\{(\mathbf{y} - \hat{\mathbf{y}})^T(\mathbf{y} - \hat{\mathbf{y}})\}. \quad (1)$$

A költségfüggvény egy előnyös megválasztása az, ha értéke a kimenetek különbségétől, vagyis a modell hibájától négyzetesen függ, ugyanis az ilyen költségfüggvény minimum-keresése matematikailag egyszerű, és emellett fizikailag a hibajel teljesítményének minimalizálását végezzük. Az olyan szűrőt, melynek együtthatóit ilyen módon definiált költségfüggvény értékének minimalizálása érdekében hangoljuk, *Wiener-szűrőnek* nevezzük.

A paraméterillesztés elvégzéséhez azonban nem elegendő a mért értékpárok pillanatnyi értékeinek ismerete, szükség van azok statisztikai jellemzőinek, vagy legalábbis bizonyos fokú momentumaiknak ismeretére. A lineáris regresszió alkalmazásakor például elegendő, ha az első- és másodfokú momentumok rendelkezésre állnak.

Ha az illesztendő modell speciálisan a paraméterek lineáris függvénye, és négyzetes költségfüggvényt alkalmazunk, a hibafüggvény a paraméterek síkja fölött kifeszített felület, melynek minimum-keresése után azonnal egyértelműen kiadódik a paraméterek optimális értéke. Ehhez viszont szükséges a hibafelület teljes ismerete.

Az illesztendő modell bemenete, kimenete és paraméterei közötti összefüggés a következő:

$$\hat{\mathbf{y}} = \hat{\mathbf{g}}(\mathbf{u}) = \hat{\mathbf{g}}(\mathbf{W}, \mathbf{u}), \quad (2)$$

ahol \mathbf{u} a bemenőjel-vektor, \mathbf{W} az állítható paraméterek mátrixa, $\hat{\mathbf{y}}$ pedig a modell kimenete. A bemenet és kimenet, valamint a modell kimenete mind-mind legalább gyengén stacionárius sztochasztikus folyamatok bizonyos realizációi. A jelölés kifejezi, hogy a modell a \mathbf{W} paramétermátrixon keresztül adaptálható.

A modell dinamikus és esetlegesen nemlineáris részét ($f(\mathbf{u}) = \mathbf{x}$) célszerű rögzíteni, és különválasztani a lineáris, adaptálható résztől (\mathbf{W}):

$$\hat{\mathbf{y}} = \hat{\mathbf{g}}(\mathbf{W}, \mathbf{u}) = \mathbf{W}^T f(\mathbf{u}) = \mathbf{W}^T \mathbf{x}, \quad (3)$$

ahol $\mathbf{x} = f(\mathbf{u})$ a modell rögzített része, amely az \mathbf{u} bemenőjel-vektorból előállítja az \mathbf{x} regressziós vektort, amely ezután az adaptálható rész bemenetét képezi. Az $\hat{\mathbf{y}}$ egyetlen kimenet esetén két vektor skaláris szorzata, több kimenet esetén viszont a képletből is látható módon mátrix-vektor szorzatként áll elő. A kifejezések alakja egy bemenet és egy kimenet esetében:

$$\hat{y} = \hat{g}(\mathbf{w}, u) = \mathbf{w}^T f(u) = \mathbf{w}^T \mathbf{x}. \quad (4)$$

A sztochasztikus folyamatokról az időfüggvényekre (a folyamatok realizációira) áttérve:

$$\hat{y}(n) = \mathbf{w}^T(n)\mathbf{x}(n) = \sum_{i=1}^{M-1} w_i(n)x_i(n), \quad (5)$$

ahol M a \mathbf{w} és \mathbf{x} vektorok hossza, a mátrix-vektor szorzat pedig vektorok skaláris szorzatává egyszerűsödik, és (5) alapján számítható.

A szétválasztás eredményeképpen az adaptálható rész paramétereiben lineáris, és a paraméterek megváltoztatása által okozott tranziens véges idő alatt kifut a rendszerből.

Mivel már felírtuk a modell kimenetének kifejezését az \mathbf{x} regressziós vektor, illetve a \mathbf{w} paramétervektor segítségével (4 és 5), írjuk fel a modellillesztéshez használt kritériumfüggvényt (1) az átlagos négyzetes hiba alapján:

$$\varepsilon(n) = E\{(y(n) - \hat{y}(n))^2\} = E\{(y(n) - \mathbf{w}^T\mathbf{x}(n))^2\} = \quad (6)$$

$$= E\{y^2(n)\} - 2 \underbrace{E\{y(n)\mathbf{x}^T(n)\}}_{\mathbf{p}^T} \mathbf{w} + \mathbf{w}^T \underbrace{E\{\mathbf{x}(n)\mathbf{x}^T(n)\}}_{\mathbf{R}} \mathbf{w} = \quad (7)$$

$$= E\{y^2(n)\} - 2\mathbf{p}^T\mathbf{w} + \mathbf{w}^T\mathbf{R}\mathbf{w}, \quad (8)$$

ahol \mathbf{p}^T a kimenet és a regressziós vektor közötti keresztkorrelációs vektor, illetve \mathbf{R} a regressziós vektor autokorrelációs mátrixa. A kritériumfüggvény (8) szerinti kifejezése egy $M + 1$ -dimenziós térben elhelyezkedő paraboloidot ír le. A kritériumfüggvény minimuma az M -dimenziós \mathbf{w} vektor szerinti differenciálással a következők adódik:

$$\frac{\partial \varepsilon}{\partial \mathbf{w}} = -2\mathbf{p} + 2\mathbf{R}\mathbf{w} = 2(\mathbf{R}\mathbf{w} - \mathbf{p}) = \mathbf{0}, \quad (9)$$

melynek megoldása a Wiener-Hopf egyenlet, mely megadja az optimum helyét a \mathbf{w} szűrőegyütthetők terében:

$$\mathbf{w}_{opt} = \mathbf{R}^{-1}\mathbf{p}. \quad (10)$$

Az optimális paraméterkészlet meghatározása ezek alapján szélsőérték-keresési feladat. Az eredmény nehézsége viszont az, hogy a \mathbf{w}_{opt} együtthetőkészlet számításához a modellezendő rendszer be- és kimeneti jeleinek statisztikai tulajdonságait *a priori* ismernünk kell a korrelációs vektorok és mátrixok meghatározásához.

A korrelációs vektorok ismeretének hiányában, illetve azok részleges ismerete esetében a paraméterbeállítás csak iteratív módon végezhető. Ekkor a statisztikai paraméterek helyett a jelek pillanatnyi értékeinek felhasználásával törekszünk az optimális paraméterkészlet megközelítésére. Az iteráció egyes lépéseiben meg kell határozni, hogy az adott pontban mennyi a hibafelület gradiense, és a következő lépés paraméterkészletét ennek ismeretében kell kialakítani.

Egy lehetséges megközelítés a legmeredekebb lejtő módszere, amelyben a legmeredekebb irányban, a negatív gradiens mentén mozdulunk el az ún. bátorsági tényező által megszabott mértékben. Ennek alkalmazásával elkerülhetjük a bonyolult számítások elvégzését, valamint kiküszöbölhetjük a statisztikai adatok előzetes ismeretének hiányát is.

2.2. Adaptív jelfeldolgozó algoritmusok

2.2.1. Az LMS algoritmus

A statisztikai jellemzők megkövetelt ismeretét kiküszöbölhetjük, valamint a bonyolult számításokat is elkerülhetjük, ha a költségfüggvényben az átlagos hiba helyett csak a pillanatnyi hibát vesszük figyelembe, és a legmeredekebb lejtő módszerét eszerint módosítjuk. Így jár el az LMS¹ algoritmus a modellparaméterek adaptálásakor.

¹Least Mean Squares

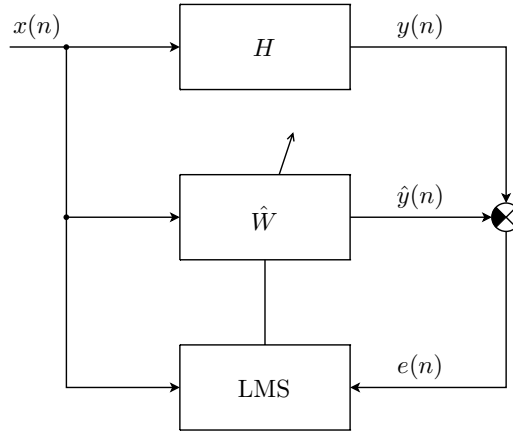
Az \mathbf{R} autokorrelációs mátrix, valamint a \mathbf{p}^T keresztkorrelációs vektor tehát a következőképpen módosul:

$$\mathbf{R} = E\{\mathbf{x}(n)\mathbf{x}^T(n)\} \text{ helyett: } \hat{\mathbf{R}} = \mathbf{x}(n)\mathbf{x}^T(n), \quad (11)$$

$$\mathbf{p}^T = E\{y(n)\mathbf{x}^T(n)\} \text{ helyett: } \hat{\mathbf{p}}^T = y(n)\mathbf{x}^T(n). \quad (12)$$

A pillanatnyi hibán alapuló becslés miatt azonban a paraméterbeállítás módosítása pontatlan lesz, nem feltétlenül a negatív gradiens irányába történik. Hosszabb időintervallumban azonban az egyes módosítások pontatlansága kiátlagolódik. A módszer alkalmazásával viszont a modell a becsült rendszer parametrikus változásait jobban képes követni. Az LMS algoritmus sajátossága, hogy a hibaminimum körül a pillanatnyi hiba kicsi, a hibafelület gradiense nulla körül van. Így a paraméterkészlet soha nem állandósul a minimális értéken, azt ugyanis nem is éri el, hanem annak egy kis környezetében lépésről-lépésre ugrál. Az LMS algoritmusban ezek alapján tehát a paramétertérben történő lépések mértékét, a *bátorsági tényezőt* célszerű kicsire megválasztani. Ezzel azonban az algoritmus beállási tulajdonságai romlanak. A μ bátorsági tényező optimális értéke az \mathbf{R} autokorrelációs mátrix ismeretében meghatározható.

Az LMS algoritmust alkalmazó legegyszerűbb struktúra blokkvázlata az 1. ábrán látható.



1. ábra. Az LMS algoritmus blokkvázlata

Az LMS algoritmusban kiszámításra kerülő rekurzív egyenletek a hibára, valamint az együtthatókra vonatkozóan a következőkre adódnak:

$$e(n) = y(n) - \hat{\mathbf{w}}^T(n)\mathbf{x}(n), \quad (13)$$

$$\hat{\mathbf{w}}(n+1) = \hat{\mathbf{w}}(n) + 2\mu e(n)\mathbf{x}(n). \quad (14)$$

Az LMS algoritmus által használt modell, melynek paraméterei hangolásra kerülnek, egy véges impulzusválaszú (FIR) digitális szűrő, melynek impulzusválasza a \mathbf{w} vektor. Az n -edik ütemben rendelkezésre álló szűrőegyüttható-készletet a $\hat{\mathbf{w}}(n)$ vektor adja meg. Az LMS algoritmus tehát az \mathbf{R} autokorrelációs mátrix, valamint a \mathbf{p}^T keresztkorrelációs vektor ismerete nélkül képes a \mathbf{w}_{opt} együtthatókészletet megközelítő szűrőegyütthatókat előállítani.

2.2.2. Az NLMS algoritmus

Ha az LMS algoritmus stabilitási és beállási tulajdonságait egyaránt megfelelően jobbra, illetve gyorsabbra kívánjuk beállítani, nyilvánvaló, hogy a két szempont szerint ellentétes irányban kellene a bátorsági tényezőt hangolni. A problémára egy lehetséges megoldást kínál az NLMS² algoritmus. Az NLMS a bátorsági tényezőt a bemeneti jel alapján normálja, így jobb stabilitási tulajdonságokkal, adott feltételek mellett általában gyorsabb

²Normalised LMS

beállással rendelkezik. Normalizálás nélkül az LMS algoritmus beállási sebessége négyzetesen függ a referencijel amplitúdójától. A normalizálás ezt kiküszöböli, azaz a konvergencia a rendszer gerjesztésének (a referenciának) az amplitúdójától független lesz.

Az NLMS algoritmus rekurzív összefüggései a hibára valamint az együtthatókra vonatkozóan a következők:

$$e(n) = y(n) - \hat{\mathbf{w}}^T(n)\mathbf{x}(n), \quad (15)$$

$$\hat{\mathbf{w}}(n+1) = \hat{\mathbf{w}}(n) + \frac{\tilde{\mu}}{a + \mathbf{x}^T\mathbf{x}}e(n)\mathbf{x}(n). \quad (16)$$

Az a regularizációs konstans funkciója az, hogy kis bemenetek esetén se válhasson a tört nevezője nullához közelivé, és így a lépésköz túl nagygyá.

2.2.3. Az FxLMS algoritmus

Egyes gyakorlati alkalmazásokban az LMS algoritmus segítségével adaptált paraméterkészletű szűrő kimenetére – a rendszer fizikai felépítéséből adódóan – egy másik, nem egységnyi átvitelű tag is kapcsolódik, és a különbségi jel képzésekor ennek kimenete kerül felhasználásra. Ezekben az esetekben tehát nem lehet már feltételezni, hogy az adaptív szűrő kimenete az identifikált rendszer kimenetéből azonnal kivonható, és az így előállított különbségi jel azonnal felhasználható az adaptív algoritmusban az adaptív szűrő hangolására.

Másképp megfogalmazva, a blokkvázlatot tekintve, az adaptív szűrő kimenete és a különbségképző blokk közé egy átvitel iktatódik, melynek legnagyobb problémája az, hogy a frekvenciatartománybeli szűrésen kívül fázistolást, késleltetést realizál az adaptív szűrő kimenete és a különbségképző között, így könnyen instabilitást okozhat, ha a fázistolás miatt nem a negatív gradiens irányában módosítjuk a szűrőegyütthatókat. Az ily módon a beavatkozó jel útjába iktatott szűrőt a – főként az aktív zajcsökkentéssel foglalkozó – szakirodalomban „másodlagos útnak” nevezik (az „elsőleges út” a modellezendő rendszert jelenti).

Ha tehát egy adott bemeneti jelre az identifikálandó rendszer valamilyen késleltetés mellett reagál, az LMS algoritmus a bemenet és a kimenet megfigyelése alapján igyekszik a szűrőt a rendszerrel azonosra hangolni, nem számol viszont azzal, hogy egy beiktatott átvitel módosítja a szűrő kimenetét. Az ún. másodlagos átvitel hatása akkor eliminálódik, ha az adaptív szűrő az identifikált rendszer mellett a másodlagos átvitel inverzét is modellezi. Szélsőséges esetben, ha a beiktatott átvitel késleltetése nagyobb az eredeti rendszer késleltetésétől, ez a feladat nem valósítható meg. Ha a másodlagos átvitel bizonyos frekvenciákon jelentős fázistolást realizál, az a szabályozási hurok instabilitásához vezet.

A probléma kiküszöbölésére több elvi megoldás is kínálkozik. Elméleti jelentőségű az az elgondolás, hogy a másodlagos átviteli úttal sorosan az inverzét elhelyezve annak hatása eliminálható. Ez a gyakorlatban nem feltétlenül valósítható meg, mert az átviteli függvény inverze nem biztos, hogy létezik. A gyakorlatban alkalmazható megoldást az FxLMS algoritmus által megvalósított struktúra kínálja. E megközelítés szerint a másodlagos átvitel becslőjét kell elhelyezni az LMS algoritmus bemenetén oly módon, hogy az a referencijeleket szűrje, de az adaptálandó szűrő bemenetére ne legyen hatással. Ez a felépítés tekinthető meg a 2. ábrán. A rendszer lineáris volta alapján a másodlagos út átvitele az adaptív szűrő elé is képzelhető, így nyilvánvalóvá válik, miért szükséges a referencijeleket is ezzel az átvittel megszüntetni. Ennek megvalósításához azonban szükséges a másodlagos átvitel megfelelő pontosságú ismerete, amelyre egy lehetséges módszer az LMS algoritmus alkalmazása.

Az FxLMS algoritmusban a hibajel a következőknek megfelelően áll elő:

$$e(n) = y(n) - \hat{y}_s(n), \quad (17)$$

ahol $\hat{y}_s(n)$ az $\hat{y}(n)$ jel S másodlagos átvittel szűrt változata. Legyen C átvitelű blokk az S átvitel becslője, valamint \mathbf{c} annak véges impulzusválasza (a vektor a FIR szűrőegyütthatókat tartalmazza)! Ekkor $e(n)$ felírható a következő módon:

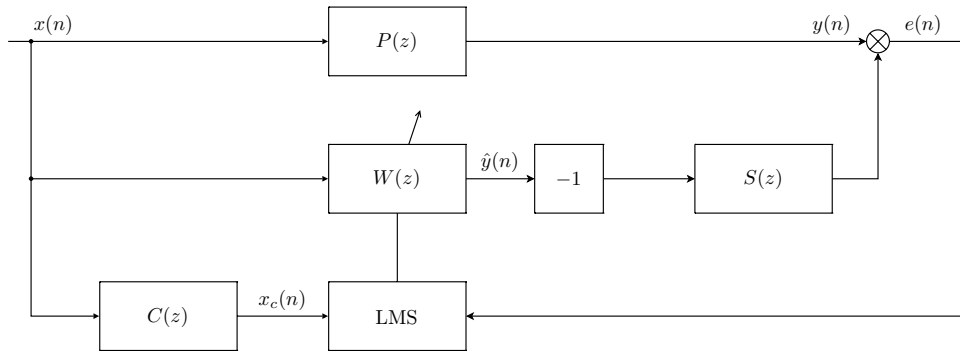
$$e(n) = y(n) - \hat{\mathbf{w}}^T(n)\mathbf{x}_c(n), \quad (18)$$

$$\mathbf{x}_c(n) = \begin{bmatrix} \sum_{i=0}^{I-1} c_i x(n-i) \\ \sum_{i=0}^{I-1} c_i x(n-i-1) \\ \dots \\ \sum_{i=0}^{I-1} c_i x(n-i-(M-1)) \end{bmatrix}. \quad (19)$$

A $\mathbf{c} = [c_1, c_2, \dots, c_I]$ vektor a másodlagos átvitel becslőjének (véges) impulzusválasza, I pedig \mathbf{c} hossza.

A szűrőegyütthatók adaptálására alkalmazható rekurzív egyenlet:

$$\hat{\mathbf{w}}(n+1) = \hat{\mathbf{w}}(n) + \mu e(n) \mathbf{x}_c(n). \quad (20)$$



2. ábra. Az FxLMS algoritmus blokkvázlata

3. Az adaptív jelfeldolgozás módszerét alkalmazó struktúrák

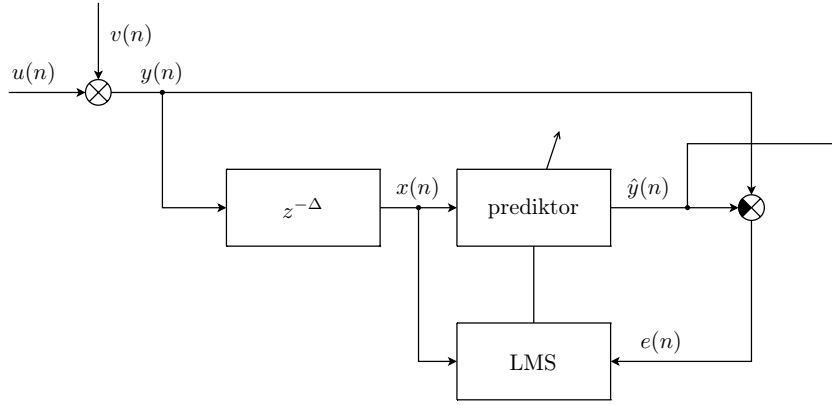
3.1. Adaptív vonaljavító

Az adaptív digitális jelfeldolgozás egyik fő területe a különféle zajok elnyomását tűzi ki célul. Egyik lehetséges felmerülő gyakori probléma például az információt hordozó jel kiszűrése az azt terhelő additív zajból. Az adaptív vonaljavító, azaz ALE³ egy ilyen feladatot megvalósító struktúra. Alkalmazására olyan esetekben lehet szükség, amikor a keskeny sávú hasznos jel sáv szélessége és egyéb paraméterei időben változnak, míg kevés előzetes, *a priori* ismeret áll rendelkezésünkre róluk. Ilyen alkalmazási terület lehet például a szonár, bizonyos orvosi alkalmazások valamint különféle beszéddel kapcsolatos feldolgozások.

Az adaptív vonaljavító egy olyan speciális zajelnyomó struktúra, amely lehetővé teszi a bemenetét terhelő zaj megfelelő mértékű csillapítását, elnyomását, míg a hasznos jelet hordozó bemeneti komponenst ideális esetben egységnyi erősítéssel, de legalábbis kis csillapítással továbbengedi. Az adaptív jelfeldolgozás miatt nem szükséges, hogy a jel stacionárius tulajdonságú legyen.

Az ALE struktúráját tekintve egy késleltető, valamint egy lineáris prediktor összekapcsolásából épül fel, amint az a 3. ábrán látható. Az ALE $y(n)$ bemeneti jele a hasznos $u(n)$ jelből, valamint az ahhoz adódó $v(n)$ zajból áll. A prediktor $\hat{y}(n)$ kimeneti jelét a $y(n)$ bemeneti jelből kivonva megkapjuk az $e(n)$ becslési hibát, melyet a prediktor adaptív hangolására használunk fel. A lineáris prediktort esetünkben egy FIR szűrő valósítja meg, melynek együtthatóit az LMS algoritmus hangolja.

³Adaptive Line Enhancer



3. ábra. Az ALE struktúra blokkvázlata

A prediktor $x(n)$ bemeneti jele a ALE bemenetének Δ számú mintával késleltetett változata:

$$x(n) = y(n - \Delta). \quad (21)$$

A struktúra Δ késleltetésének mértékét a $v(n)$ zajkomponensnek megfelelően kell megválasztani. Amennyiben a zajkomponens szélessávú, a késleltetést olyan nagyra kell választani, hogy $v(n)$, valamint annak $v(n - \Delta) \dots v(n - \Delta - M)$ késleltetett változata korrelálatlan legyen (M a FIR szűrő hosszát jelöli). Ily módon a becslési hiba képzésekor a bemenet zajkomponense, valamint az adaptív szűrő kimenetének a bejövő zajból származó komponense korrelálatlan lesz. Ennek hatására a szűrő hangolásakor nem fordul elő az, hogy az adaptív algoritmus a szűrőegyütthatókat olyan irányba állítja, hogy a szűrő a zajt is átengedje annak érdekében, hogy a különbségképzéskor a becslési hibát még jobban minimalizálhassa. Ekkor a struktúra kimeneti jele az $\hat{y}(n)$ jel. A Δ késleltetést túl nagyra választva azonban a hasznos (nemstacionárius) jel mintái is korrelálatlanok lesznek, és a struktúra nem működik.

Amennyiben a zajkomponens keskenysávú, a zaj és annak igen nagy késleltetéssel rendelkezésre álló mintái is korreláltak (periodikus gerjesztés esetén tetszőleges késleltetés esetén is). Ebben az esetben a késleltetést olyan nagyra kell választani, hogy $u(n)$, valamint annak $u(n - \Delta) \dots u(n - \Delta - M)$ késleltetett változata korrelálatlan legyen (M a FIR szűrő hosszát jelöli). Ily módon a becslési hiba képzésekor a bemeneti jel, valamint az adaptív szűrő kimenetének a hasznos jelből származó komponense korrelálatlan lesz. Ekkor viszont a struktúra kimeneti jele az $e(n)$ jel.

Ha tehát a Δ ún. dekorreláló késleltetés értéke megfelelő, lehetővé válik a zaj kioltása, ugyanakkor a késleltetés és a kimenet megválasztása az elnyomandó zaj tulajdonságaitól függ.

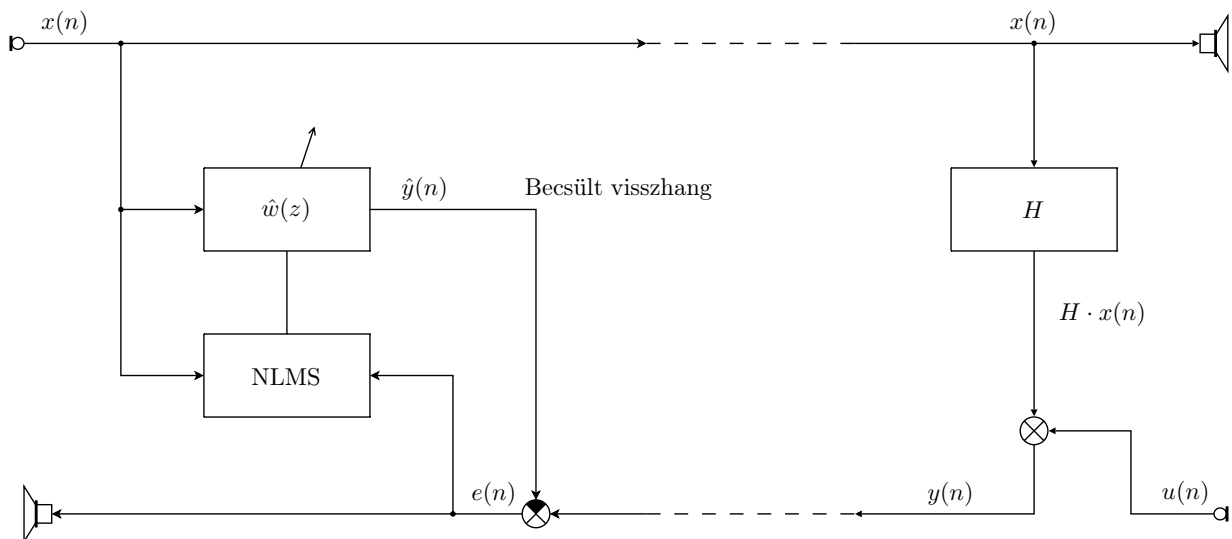
3.2. Adaptív visszhangcsökkentés

A visszhang problémája legszélesebb körben elsőként a PSTN⁴ telefonhálózatokban jelentkezett. A kapcsolóközpontokban elhelyezett ún. hibridek közötti nagy távolság és a tökéletlen illesztés miatt reflexió lép fel, amely a nagy késleltetés miatt hallható visszhangot képes okozni, amely bizonyos határ fölött zavarja a beszélőket, vagy akár a beszédérthetőséget is jelentősen ronthatja.

Az adaptív visszhangcsökkentő struktúra modellje a 4. ábrán látható. Az ábra bal oldalán látható hangszóró-mikrofon pár az egyik, míg a jobb oldalon látható pár a másik személyhez tartozó végpont. A szaggatott vonalak a hosszú átviteli utat szemléltetik, de a visszhang szempontjából az ezek által okozott késleltetést beleértjük a H átvitelbe, amely a hallgató oldalán elhelyezkedő hibrid átvitelét jelöli.

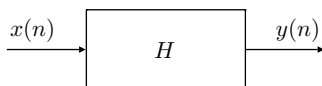
A beszélő akkor hallja meg saját visszhangját, ha $H \neq 0$. Tekintsük azt az esetet, amikor a hallgató csendben van (ekkor $u(n) = 0$). Ebben az esetben a hallgató oldaláról $y(n) = H \cdot x(n)$ visszhang indul vissza a

⁴Public Switched Telephone Network



4. ábra. Az adaptív visszhangcsökkentő struktúra blokkvázlata

beszélő felé. Az 5. ábrán a 4. ábra jobb oldalát láthatjuk erre az esetre vonatkozóan újrarajzolva. Ebből már látható, hogy az adaptív visszhangcsökkentés nem más, mint a hallgató oldalán elhelyezkedő hibrid átvitelének rendszeridentifikációs feladata.



5. ábra. Ha $u(n) = 0$, a hallgatóoldali hibridet identifkáljuk

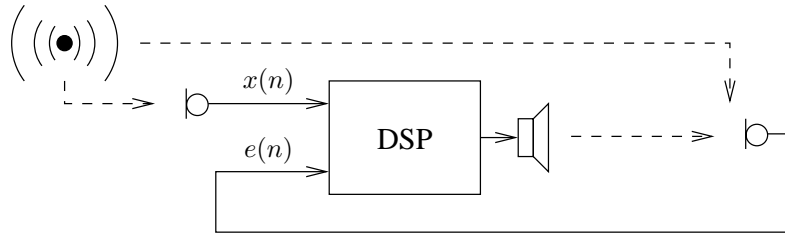
Ha a visszhangot alkalmas módon becsülni tudjuk, az $\hat{y}(n)$ becsült visszhangjelet kivonhatjuk a beszélő oldalára visszaérkező $y(n)$ jelből. A két jel különbsége a becslési hibajel: $e(n)$. A becslés az $e(n)$ hibajel és a beszélő $x(n)$ jelét felhasználó, a 2.2.2. pontban ismertetett NLMS algoritmussal történik. Fontos, hogy az algoritmus számára szükséges minden jel egy helyen rendelkezésre áll, jelen esetben az ábra bal oldalán.

3.3. Aktív zajcsökkentés

Az aktív zajcsökkentés feladata a környezetből, vagy valamilyen zajforrásból érkező zajnak ellenfázisú zajjal történő kioltása. Szép, önálló szakterületet is teremtő alkalmazása ennek az akusztikus zajok aktív elnyomása. Ez esetben arra alkalmas helyeken ún. beavatkozó hangszórók elhelyezésére van szükség, és a segítségükkel kiadott ellenfázisú zajjal a hibamikrofonok bizonyos környezetében csendes zónák alakíthatók ki. Az akusztikai rendszerek nagy dinamikatarományban jó közelítéssel lineárisnak tekinthetők, így a szuperpozíció elve eredményesen alkalmazható. Az aktív zajcsökkentés az akusztikai alkalmazásokon kívül mechanikai rezgések csillapítására is eredményesen alkalmazható, a megfelelő szenzorok és beavatkozók ebben az esetben is rendelkezésre állnak (például gyorsulásérzékelők és rázóasztalok).

A 6. ábrán egy aktív zajcsökkentő rendszer blokkvázlata látható, amely akusztikus zaj kioltását végzi. Az ábra bal oldalán található a zajforrás, amelynek hangja a levegőben terjed, és amelyet a DSP-hez közvetlenül kapcsolódó ún. referenciamikrofon, valamint a védendő térrészben elhelyezett ún. hibamikrofon vesz. A zajforrástól a hibamikrofonig terjedő átvitel az elsődleges út (*primary path*), míg a beavatkozó hangszórótól a hibamikrofonig terjedő átvitel a másodlagos út (*secondary path*), a szakirodalomban alkalmazott terminológiát követve.

A fizikai elrendezést reprezentáló ábra összevethető az elvi blokkvázlattal, amely az FxLMS algoritmust bemutató 2. ábrán látható. Az $x(n)$ referenciajel és az $y(n)$ elnyomandó zaj közötti átvitel P . (Itt a zajforrás és



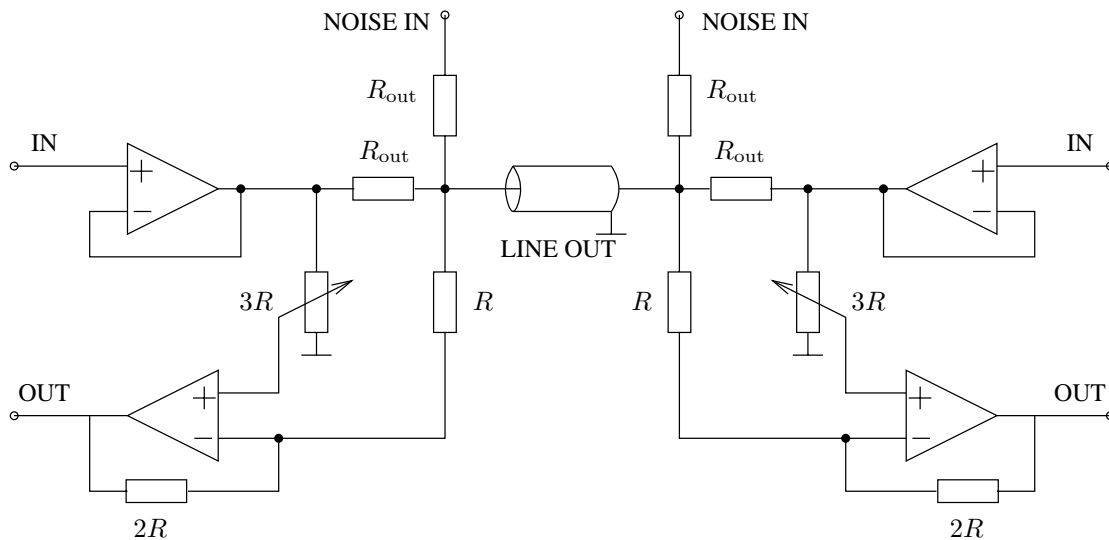
6. ábra. Aktív zajcsökkentő rendszer

a referenciajel közötti átvitelt konstansnak tekintjük.) Az S átvitel az adaptív szűrő $\hat{y}(n)$ kimeneti jele és $e(n)$ hibajel közötti átvitelt jelenti. A fázisfordításra azért van szükség, mert a fizikai rendszerben a szuperpozíció csak összegzést végez. S nemcsak az akusztikus átvitelt reprezentálja, hanem a jelútban lévő összes eszközt (erősítők, AD- és DA-átalakítók stb.) is. A jelek összegzése nem a DSP-ben valósul meg, és közvetlenül nem is hozzáférhető. Éppen ezért S nem lehet egységnyi, így adaptációra mindenképpen az FxLMS algoritmust kell alkalmazni.

A másodlagos út identifikálása az LMS algoritmussal történhet. Az identifikálandó rendszer bemenete a hangszóróra adandó jel, kimenete pedig hibamikrofon jele. Megfelelő gerjesztés (pl. fehér zaj) esetén, beállítás után az adaptív szűrő \hat{W} átvitelt reprezentáló együtthatóit át kell másolni C együtthatóinak helyére, ezáltal C valóban az S másodlagos út becslője lesz.

4. A mérendő objektum

A mérés során egy elektronikus hibrid-páron kell kísérleteket végezni. Az áramkör egyszerűsített kapcsolási rajza a 7. ábrán látható. Az áramkörben az IN bemenetre adott jel kerül továbbításra a LINE OUT vonali kimenetre,



7. ábra. A mérendő elektronikus hibrid áramkör.

a LINE OUT felől érkező jel pedig az OUT kimeneten férhető hozzá. A NOISE IN bemeneten lehet a csatornához zajt adni. A hibrid pár két oldalát koaxiális kábel segítségével lehet összekötni. Az áramkörben $R_{out} \ll R$. Ha a potenciométert úgy állítjuk be, hogy a vonalat meghajtó műveleti erősítő feszültségét harmadára ossza le, akkor az áramkör az azonos oldali IN bemenetre adott jelet elnyomja, míg a vonalon érkező jelet egységnyi erősítéssel viszi át. (Ennek belátását az Olvasóra bízjuk.) Az áramköri elemek tűrése, valamint a fizikai körülmények megváltozása miatt ez azonban nem teljesen pontosan, továbbá a valóságban a vevő oldalon akusztikus csatolás

léphet fel a kimenet és bemenet között, ezért szükséges az adaptív visszhangcsökkentés, amelyet az egyik oldalon kell DSP kártya segítségével megvalósítani.

A NOISE IN bemenetre adott zajt kompenzálni lehet. Egyrészt lehetőség van arra, hogy az OUT kimeneten adaptív vonaljavítót működtessünk, másrészt az IN bemenetre adott zajjal aktív zajcsökkentést valósíthatunk meg. Ekkor a hibajel az OUT kimeneten jelenik meg. Ebben az esetben a másodlagos út dinamikáját döntően az AD- és DA-átalakítók alkotják.

5. A jelfeldolgozó kártya és a fejlesztői környezet

5.1. A jelfeldolgozó processzorok

A DSP-k⁵, azaz a digitális jelfeldolgozó processzorok széles körben elterjedtek a beágyazott rendszerek területén. Megtalálhatók mobiltelefonokban, hang- és video, DVD lejátszóknak, digitális fényképezőgépekben és kamerákban, valamint számos olyan ipari alkalmazásban, ahol jelekkel vagy adatfolyamokkal kell dolgozni. Az általános célú processzorokkal szemben a DSP-k valós idejű jelfeldolgozási alkalmazásokra kerülnek optimalizálásra mind architektúráis szinten, mind az utasításkészletük kialakítása terén. Úgy alakítják ki őket, hogy a hatékony jelfeldolgozást a lehető legkevesebb utasítás végrehajtása mellett lehessen használatukkal elvégezni.

5.2. Az ADSP-21364 EZ-KIT Lite fejlesztőkártya

Az ADSP-21364 EZ-KIT Lite fejlesztőkártya az ADSP-21364 jelfeldolgozó processzorral történő fejlesztés hatékony és gyors eszköze. A fejlesztőkártyán kialakításra került a jelfeldolgozó processzor minden olyan szükséges perifériája és környezete, amelyre a leggyakoribb feladatok elvégzésekor szükség lehet. Az analóg és digitális hang ki- és bemenetek, LED-ek, kapcsolók, nyomógombok azonnal hozzáférhetőek, így a szoftverfejlesztés mindenféle hardvertervezési lépés nélkül azonnal elkezdhető. A fejlesztőkártya tervezésekor igyekeztek a lehető legtöbb felhasználási lehetőséget kipróbálhatóvá, a processzor által nyújtott funkcionalitást pedig a lehető legmélyebben megismerhetővé tenni. A kártya a következő hardverelemeket bocsátja a fejlesztő rendelkezésére:

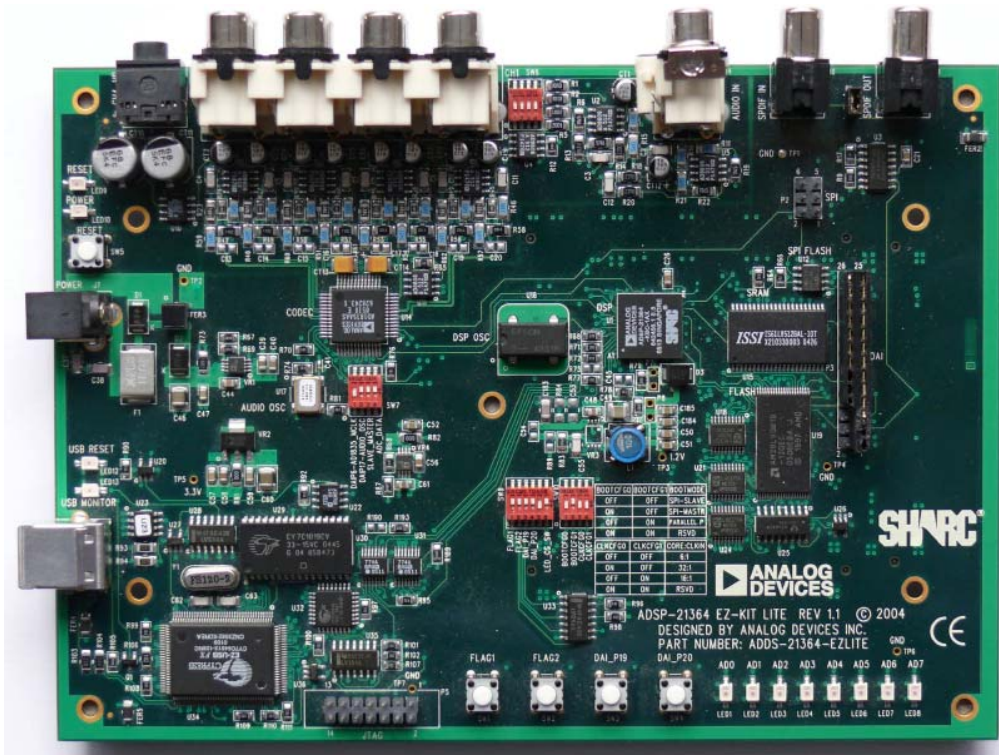
- Analog Devices ADSP-21364 processzor
- 512 kbit \times 8 bit SRAM⁶
- 1 Mbit \times 8 bit flash memória
- 2 Mbit SPI⁷ által kezelhető flash memória
- Analóg hanginterfész (AD1835A kodek, 1 db sztereó bemenet, 4 db sztereó kimenet)
- Digitális hanginterfész (1 db bemenet, 1 db kimenet)
- 11 db LED (ebből 1 db „power”, 1 db „board reset”, 1 db „USB monitor”, 8 db általános célú)
- 5 db nyomógomb (1 db „reset”, 2 db DAI lábra, 2 db FLAG lábra kötve)
- Bővítőinterfész (párhuzamos port, FLAG-ek, DAI, SPI)
- JTAG emulátor port
- USB port a PC csatlakoztatásához.

A kártya képe a 8. ábrán, a csatlakozók oldalnézeti képe pedig a 9. ábrán látható.

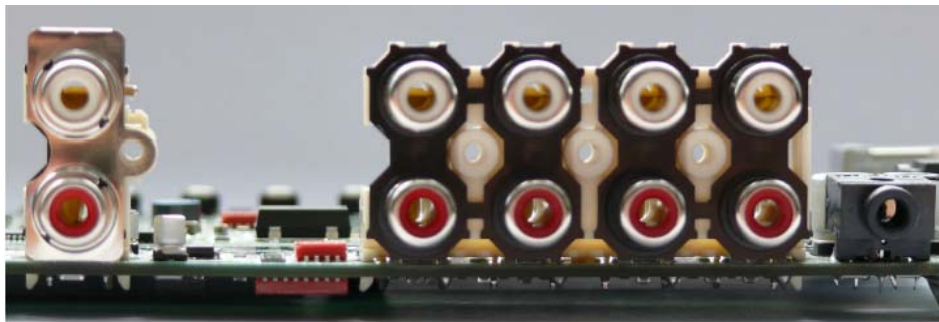
⁵Digital Signal Processor

⁶Synchronous Random Access Memory

⁷Serial Peripheral Interface



8. ábra. Az ADSP-21364 jelfeldolgozó kártya képe



9. ábra. Az ADSP-21364 jelfeldolgozó kártya csatlakozóinak képe

5.3. A Visual DSP++ v4.5 fejlesztői környezet

A fejlesztői környezetet több laboratóriumi méréshez is használjuk, ezért ismertetése külön dokumentumban található. Jelen mérés során igazi fejlesztési feladatot nem kell megoldani, előre elkészített programokat kell futtatni, illetve a paraméterek megváltoztatása érdekében a memória tartalmát módosítani.

6. A mérési feladatok

6.1. Szimulációk MATLAB-ban

6.1.1. Az LMS algoritmus megvalósítása

1. Az elméleti áttekintésben szereplő (13) és (14) egyenleteknek megfelelően írjon MATLAB függvényt az LMS algoritmus 1. ábra szerinti rendszeridentifikációs feladatot ellátó megvalósítására! Az algoritmust megvalósító függvény váza a következő legyen:

```

function [e,w]=lms(mu,M,x,y);
%bemeneti paraméterek:
%mu: lépésköz (bátorsági tényező)  $1 \times 1$ 
%M: a szűrő együtthatóinak száma  $1 \times 1$ 
%x: a szűrő bemenete  $N \times 1$ 
%y: a szűrő kimenete  $N \times 1$ 

%kimeneti változók:
%e: becslési hiba  $N \times 1$ 
%w: szűrőegyütthatók  $M \times 1$ 

```

2. Tervezzon IIR szűrőt a mérésvezető utasításai szerint!
3. Identifikálja az IIR szűrőt az LMS algoritmus segítségével! Gerjesztésként használjon Gauss-eloszlású fehér zajt! Vesse össze az IIR szűrő és az identifikáció eredményeként kiadódó FIR szűrő átvitelét!
4. Próbálja ki különféle μ és M értékeket, és vizsgálja meg, hogyan befolyásolják az adaptív szűrő konvergenciáját, illetve az IIR szűrő approximációjának jóságát!

6.1.2. Az NLMS algoritmus megvalósítása

1. Az elméleti áttekintésben szereplő (15) és (16) egyenleteknek megfelelően írjon MATLAB függvényt az NLMS algoritmus 1. ábra szerinti rendszeridentifikációs feladatot ellátó megvalósítására! Az algoritmust megvalósító függvény váza a következő legyen:

```

function [e,w]=nlms(mu,a,M,x,y);
%bemeneti paraméterek:
%mu: lépésköz (bátorsági tényező)  $1 \times 1$ 
%M: a szűrő együtthatóinak száma  $1 \times 1$ 
%x: a szűrő bemenete  $N \times 1$ 
%y: a szűrő kimenete  $N \times 1$ 
%a: regularizáló konstans  $1 \times 1$ 

%kimeneti változók:
%e: becslési hiba  $N \times 1$ 
%w: szűrőegyütthatók  $M \times 1$ 

```

2. Próbálja ki különféle gerjesztőjel-teljesítményt, illetve μ értékeket, és vizsgálja meg, hogyan befolyásolják az adaptív szűrő konvergenciáját! Az előző feladatban tervezett IIR szűrőt identifikálja, és az eredményeket vesse össze az ott kapottakkal!

6.1.3. Az FxLMS algoritmus megvalósítása

1. Az elméleti áttekintésben szereplő (17), (18), (19) és (20) egyenleteknek megfelelően írjon MATLAB függvényt az FxLMS algoritmus 2. ábra szerinti megvalósítására! Az S átvitel egy erre a feladatra tervezett egyszerű FIR szűrő legyen! (Mivel ez utóbbi szűrő is része az adaptációs huroknak, $\hat{y}_s(n)$ előállítására nem történhet blokkosan, azaz a korábbi lms függvény közvetlenül nem használható fel.) Az algoritmust megvalósító függvény váza a következő legyen:

```

function [e,w]=fxlms(mu,M,s,c,x,y);
%bemeneti paraméterek:

```

```

%mu: lépésköz (bátorsági tényező)  $1 \times 1$ 
%M: a szűrő együtthatóinak száma  $1 \times 1$ 
%x: a szűrő bemenete  $N \times 1$ 
%y: a szűrő kimenete  $N \times 1$ 
%s: a másodlagos út (FIR szűrő) együtthatói  $S \times 1$ 
%c: a másodlagos út (FIR szűrő) modelljének együtthatói  $C \times 1$ 

%kimeneti változók:
%e: becslési hiba  $N \times 1$ 
%w: szűrőegyütthatók  $M \times 1$ 

```

2. Identifikálja az S átvitelt az előzőleg megírt LMS algoritmussal, az identifikált együtthatókat használja fel a C szűrőben!
3. Tesztelje az FxLMS algoritmust többféle referenciajellel (szinuszjel, háromszögjel stb.)! A P átvitel legyen az előzőekben használt IIR szűrő!
4. Készítsen szimulációt, amely demonstrálja, hogy S elégtelen identifikációja esetén az algoritmus nem konvergens, még μ minden határon túl történő csökkentése esetén sem!

6.2. Kísérletek jelfeldolgozó processzorral

A feladatok megoldásához a fent ismertetett ADSP-21364 EZ-KIT Lite fejlesztőkártyát, valamint az elektronikus hibrid áramkört kell felhasználni. A kimeneti jeleket hangszórón vagy fejhallgatón célszerű meghallgatni, oszcilloszkópon pedig vizuálisan elemezni.

6.2.1. Adaptív vonaljavítás

A feladat megoldásához szükség van „hasznos”, illetve „zajt” értelemben használható forrásra. A hasznos jel legyen audio jel (rádió, MP3 lejátszó stb. kimenőjele), a zajt függvény- és zajgenerátor segítségével lehet előállítani.

A DSP kártyára a MScMeres_ALE projektet kell letölteni. A 3. ábrán látható jelöléseket alkalmazva, a mérés során az ALE struktúra $y(n)$ bemeneti jelét a jelfeldolgozó kártya bemenetének bal (pirossal jelölt) csatornájára kell kötni. A kártya kimenetein az $y(n)$, $\hat{y}(n)$, valamint az $e(n)$ jelek hallgathatók meg, illetve dolgozhatók fel:

- y : 1. kimenet bal és jobb csatornája
- \hat{y} : 2. kimenet bal és jobb csatornája
- e : 3. kimenet bal és jobb csatornája
- x : 4. kimenet bal és jobb csatornája.

1. A hibrid segítségével összegezze a hasznos jelet és a zajt! Erre többféle lehetőség is van, ügyeljen arra, hogy se az áramkört, se a forrást ne terhelje jelentősen!
2. Periodikus zajt alkalmazva működtesse a struktúrát! Vizsgálja meg, milyen Δ késleltetésre van szükség!
3. Véletlenszerű zajt alkalmazva működtesse a struktúrát! Vizsgálja meg, milyen Δ késleltetésre van szükség!
4. Próbáljon ki különféle μ értékeket, és vizsgálja a rendszer beállási és stabilitási tulajdonságait! Milyen hatással van μ változtatása a hasznos jelre?

6.2.2. Adaptív visszhangcsökkentés

A feladat megoldásához két jelforrásra van szükség. Az egyik forrás mindenképpen az előző feladatban alkalmazott audio jel legyen, a másik valamilyen felismerhető hangfrekvenciás jel legyen, de egyéb audio jel is megfelelő.

A DSP kártyára a `MScMeres_LMS` projektet kell letölteni. Az 1. ábrán látható jelöléseket alkalmazva, a mérés során az LMS struktúra $y(n)$ bemeneti jelét a jelfeldolgozó kártya bemenetének bal (pirossal jelölt) csatornájára, az $x(n)$ referenciajelet a jobb (fehérrel jelölt) csatornájára kell kötni. A kártya kimenetein az $y(n)$, $\hat{y}(n)$, valamint az $e(n)$ jelek hallgathatók meg, illetve dolgozhatók fel:

- y : 1. kimenet bal és jobb csatornája
 - \hat{y} : 2. kimenet bal és jobb csatornája
 - e : 3. kimenet bal és jobb csatornája
 - x : 4. kimenet bal és jobb csatornája.
1. Állítsa össze az adaptív visszhangcsökkentő rendszert, a potenciométert állítsa el a teljes kioltást megvalósító állásából! Tesztelje az algoritmus működését!
 2. Próbáljon ki különféle μ értékeket, és vizsgálja a rendszer beállási és stabilitási tulajdonságait! Milyen hatással van μ változtatása a hasznos jelre?

6.2.3. Aktív zajcsökkentés

A feladat megoldásához csak egy forrásra van szükség.

A DSP kártyára a `MScMeres_ANC` projektet kell letölteni. A programnak két üzemmódja van: azonosítás és zajcsökkentés. Azonosítás módban a másodlagos út azonosítása történik meg, a fehér zaj gerjesztést a DSP kártya állítja elő. Ha az algoritmus hibája kellően kicsire csökkent, át lehet váltani zajcsökkentés módba.

- Bemenetek:
 - 1. bemenet (fehér): referenciajel
ANC módban: x
Ident módban nincs használatban
 - 2. bemenet (piros): hibaérzékelő jele
ANC módban: e
Ident módban: y .
- Kimenetek:
 - 1. kimenet mindkét csatornája: beavatkozójel
ANC módban: $-\hat{y}$
Ident módban: fehér zaj
 - 2. kimenet mindkét csatornája:
ANC módban: hibajel (e)
Ident módban nincs használatban.

Az azonosítási mód a `FLAG1 / SW1` gomb nyomvatartásával működtethető. Azonosítási módban a beavatkozó kimeneten a jelfeldolgozó kártyán generált álvéletlen zaj jelenik meg. Aktív zajcsökkentés üzemmódban a `FLAG2 / SW2` nyomógomb megnyomásával a \hat{W} szűrő, míg a `DAI_P19 / SW3` nyomógombbal a C szűrő együtthatói nullázhatók. Az előbbi használatával a rendszer beállási tulajdonságai vizsgálhatók, míg a másodikkal

a referenciajel vehető le az LMS algoritmus bemenetéről azáltal, hogy csupa nulla együtthatójú szűrővel kerül megszűrésre. Referenciajel hiányában W együtthatói nem módosulnak.

A DAI_P20 / SW4 nyomógomb segítségével a kimenet tiltható. A szűrőegyütthatók stabilitása megőrzésének érdekében letiltott kimenet esetén W együtthatóinak adaptálása is felfüggesztésre kerül.

A zavarjelet jelen esetben a hibrid NOISE IN bemenetére kell adni, amelyet a IN bemenetre adott beavatkozójellel kell elnyomni. Az OUT kimenet használható a hiba (a zaj és a beavatkozás eredménye szuperpozíciója) érzékelésére. Ehhez a hibrid kiegyenlítését meg kell szüntetni.

1. Identifikálja a másodlagos utat! A blokkvázlat alapján határozza meg, milyen elemek vannak a jelútban, és az identifikáció eredményét vesse össze ezzel! (Használja a memória grafikus megjelenítését!)
2. Periodikus zavarjelet alkalmazva tesztelje az algoritmust!
3. Milyen hatása van az elnyomás mértékére, illetve a beállási, stabilitási tulajdonságokra W , illetve C fokszáma, valamint μ értéke változtatásának?