

Budapesti Műszaki és Gazdaságtudományi Egyetem
Villamosmérnöki és Informatikai Kar
Méréstechnika és Információs Rendszerek Tanszék

Nemlineáris rendszerek identifikációja gyakorlati megközelítésben

TDK DOLGOZAT

Készítette
Czupy Gergely János

Konzulens
Dr. Sujbert László

Tartalomjegyzék

1. Bevezetés	1
2. Nemlineáris rendszerek modellezése	3
2.1. Az inverz jelentősége és meghatározásának problémái	3
2.2. Az inverz online és offline meghatározása	4
2.3. Nemlineáris rendszerek modelljei és jellemzőik	5
2.3.1. Nemlineáris rendszerek definíciója és modellezésének lépései	5
2.3.2. Nemlineáris modellek csoportosítása	6
2.3.3. Hogyan válasszunk a modellek közül?	10
3. A Volterra-modell	12
4. Blokkalapú modellek és a Hammerstein-modell	14
4.1. Néhány lehetőség	15
4.2. A Hammerstein-modellről részletesen	16
4.2.1. A memóriamentes nemlinearitás modellezése	17
4.2.2. Spline-függvények	18
4.2.3. A lineáris blokk modellezése	21
5. Kiegészítő eszközök nemlineáris rendszerek kezeléséhez	22
5.1. „Mennyire nemlineáris?”	22
5.1.1. Egyenestől vett minimális, átlagos négyzetes eltérés	22
5.1.2. Itakura-Saito távolság	23
5.1.3. Torzítási tényező	23
5.1.4. Koherenciafüggvény	23
5.2. Gyakori nemlineáris karakterisztikák	23
5.2.1. Gyors becslő módszer gyengén nemlineáris, monoton növekvő karakterisztikák felmérésére	25
5.3. Algoritmusok	26
5.3.1. Az LMS algoritmus	27
5.3.2. Az NLMS algoritmus és változatai	28
5.3.3. Az RLS algoritmus	29
5.3.4. Az ELS algoritmus	30
5.4. A paraméterek illesztése	31
5.4.1. Az XLMS algoritmus és kompenzálásra használható módosítása	32
5.4.2. Az ELMS algoritmus és kompenzálásra használható módosítása	33
5.4.3. Paraméterek illesztése Hammerstein-modellhez	34

5.4.4.	Inverz meghatározása frekvencia-mintavételi eljárással	37
5.4.5.	Inverz közelítő meghatározása analitikusan	38
5.4.6.	Inverz meghatározása adaptív módszerrel	39
6.	Gyakorlati tapasztalatok – az egyszerűsített Volterra-modell	41
6.1.	A bemenő jel szűrése	41
6.2.	Együtthatók számának megválasztása	44
6.3.	Bátorsági tényező megválasztása	46
6.4.	Kompenzálás a Volterra-moddal	46
6.4.1.	Egy shaker modellezése és kompenzálása	47
7.	Gyakorlati tapasztalatok a Hammerstein-moddal	53
7.1.	Modellezés a Hammerstein-moddal	53
7.1.1.	Hammerstein-modell alapú rendszer identifikálása Hammerstein-moddal .	53
7.1.2.	Wiener-modell alapú rendszer identifikálása Hammerstein-moddal	54
7.1.3.	Shaker modellezése Hammerstein-moddal	55
7.2.	A kontrollpontok és az együtthatók számának megválasztása	56
7.3.	A bemenő jel szűrése	56
7.4.	Offline kompenzálás a Hammerstein-moddal	58
8.	Összefoglalás, kitekintés	61
A.	melléklet	68
B.	melléklet	69

Ábrák jegyzéke

1.	Szakasz és szabályzó kompenzáláshoz	3
2.	Nemlineáris rendszer kompenzálása offline meghatározott inverzzel	4
3.	Nemlineáris rendszer kompenzálása online meghatározott inverzzel	4
4.	Rendszer identifikálásának struktúrája	5
5.	Nemlineáris modellek csoportosítása Unbehauen alapján	7
6.	Nemlineáris modellek csoportosítása Ljung alapján	9
7.	Egyszerűsített Volterra-modell blokkvázlata	13
8.	Hammerstein-modell blokkvázlata	15
9.	A Wiener-modell blokkvázlata	15
10.	Wiener-Hammerstein modell	15
11.	Párhuzamosan kapcsolt Wiener-Hammerstein-modellek	16
12.	Visszacsatolt blokk-alapú modell	16
13.	Hammerstein-modell szűrt zajjal	16
14.	Wiener-modell folyamatot és mérést terhelő zajjal.	17
15.	Catmull-Rom spline-függvény	19
16.	Spline függvény és kontrollpontjai	20
17.	Gyakori nemlineáris karakterisztikák	24
18.	Hídkapcsolás	25
19.	Shaker nemlineáris karakterisztikájának közelítő becslése	26
20.	Modellillesztő technikák csoportosítása	32
21.	Az XLMS algoritmus	33
22.	Az XLMS algoritmus kompenzálásra használható módosítása	33
23.	Az ELMS algoritmus	34
24.	Az ELMS algoritmus kompenzálásra használható módosítása	34
25.	Struktúra Hammerstein-modell illesztéséhez	35
26.	Hammerstein-modell helytelen visszacsatolása	36
27.	Hammerstein-modell hibajelének alakulása helyes és helytelen visszacsatolás esetén	37
28.	Invertálható függvény adaptív invertálása	38
29.	A $H(z)$ rendszer pólus-zérus térképe	39
30.	Késleltetett inverz adaptív meghatározása	40
31.	Volterra-modell hibajelének mintavételi frekvenciától függő alakulása RLS algoritmus esetén	42
32.	Szimuláció elrendezésének blokkvázlata	42
33.	Volterra-modell hibajelének alakulása különféle hatványok és együtthatószám esetén, LMS algoritmussal	45
34.	A bátorsági tényező hatása a konvergencia sebességére Volterra-modellnél	47

35.	XLMS algoritmus offline modelljének és az online adaptált kontrollerének hibajele	48
36.	Késleltetett gerjesztés, Volterra-moddal kompenzált és kompenzálatlan kimenet arctan karakterisztikánál, 1.5 kHz-es felső sávhatárú zaj gerjesztésnél	48
37.	Volterra-moddal kompenzált és kompenzálatlan rendszer kimenetének koherenciafüggvénye arctan-szerű karakterisztikánál és szűrt bemenő zaj gerjesztésnél . .	49
38.	Shaker gerjesztése és kimeneti jele	49
39.	A shaker ugrásválasza	50
40.	Elrendezés shaker jeleinek méréséhez.	50
41.	Kontroller hibajele shaker kompenzálásakor	51
42.	Shaker késleltetett gerjesztése, kompenzált és nem kompenzált kimenete	51
43.	A shaker Volterra-moddal kompenzált és nem kompenzált kimenetének koherenciafüggvénye	52
44.	Diódát tartalmazó rendszer identifikálása Hammerstein-moddal	54
45.	Holtsávot tartalmazó Hammerstein-modell alapú rendszer identifikálása Hammerstein-moddal	55
46.	Wiener-modell alapú, arctan függvényt tartalmazó rendszer Hammerstein-moddal történő identifikálásának eredménye	56
47.	Shaker-identifikálása Hammerstein-moddal	57
48.	Az együtthatók és a kontrollpontok számának hatása a Hammerstein-modell hibajelére	57
49.	Bemenő jel szűrésének hatása a Hammerstein-modell teljesítményére	58
50.	Identifikált spline függvény és inverze	59
51.	Hammerstein-moddal kompenzált és nem kompenzált kimenet Hammerstein-alapú rendszer és arctan karakterisztika esetén	60
52.	Hammerstein-moddal kompenzált arctan alapú rendszer kompenzált és nem kompenzált kimeneti jelének koherenciafüggvénye	60
53.	Shaker DSP kártyával történő kompenzálása	62
54.	Volterra-modell hibajelének alakulása LMS algoritmus és különféle mintavételi frekvenciák esetén	68
55.	Együtthatók számának hatása a Volterra-modell hibájára, RLS algoritmus esetén	69

Tartalmi összefoglaló

Egy lineáris, dinamikus rendszer identifikálásához és irányításához bőséges eszköztár áll rendelkezésre, mivel az elmúlt évtizedekben a lineáris modellalkotás területét alaposan feltérképezték. A legtöbb probléma hatékonyan megoldható az általánosan ismert módszerekkel, melyek általában az impulzusválasz meghatározására vezethetők vissza.

A valóságban azonban a rendszerek többsége – legalább kis mértékben nemlineáris. Az alkalmazás megengedheti a torzítás elhanyagolását, bár ekkor egyedül a munkapont kis környezete biztosítja a lineárishoz közeli kapcsolatot a be- és kimeneti jel között. A nemlinearitást azonban figyelembe kell venni akkor, amikor azt a precizitás, vagy a nemlinearitás mértéke megköveteli. Ilyen alkalmazási területet jelenthetnek a diódák, nagyteljesítményű erősítők és különféle szenzorok is. Nemlineáris rendszerek modellezésére és irányítására azonban nincs egyetlen jól bevált technika. A dolgozat célja, hogy olyan megoldásokat mutasson be, melyekkel egy gyengén nemlineáris, dinamikus rendszert linearizálni lehet, vagyis meghatározni annak inverzét és így minimalizálni a be- és kimeneti jel közötti torzítást.

A dolgozat összefoglalja a releváns szakirodalomban fellelhető módszereket, tárgyalja a fontosabb algoritmusokat, linearizáló struktúrákat és nemlineáris karakterisztikákat. Külön figyelem irányul két modellre – a kernel alapú Volterra-, és a blokk-alapú Hammerstein-modellre. A dolgozat MATLAB szimulációk és egy valós alkalmazás, egy Brüel & Kjaer Mini-Shaker 4810 shaker segítségével vizsgálja a gyengén nemlineáris rendszerek modellezését és irányításának lehetőségeit – esetenként külön kitérve annak kihívásaira, hogyan lehet a gyakorlati életbe hatékonyan átültetni a technikákat.

Kulcsszavak: linearizálás, nemlineáris rendszeridentifikáció, Hammerstein-modell, Volterra-modell, késleltetett inverz

Abstract

Nonlinear system identification – a practical approach

There are several methods to model and control linear dynamic systems thanks to an intensive research during the last decades. Most of these problems can be investigated with standardized tools and algorithms that determine the system's impulse response. In the real world, however, the majority of the systems represent – at least – weakly non-linear characteristics. Also, there isn't any common approach to handle non-linear behavior. Precision and degree of non-linearity must be tackled in practice for many components such as diodes, power amplifiers, and sensors.

The primary aim of this work is to find and deeply examine methods that can be used for the identification of the inverse of a weakly non-linear dynamic system, which must be performed to minimize the distortion between the input and output signals.

The work gives an overview of the available literature, most important algorithms, structures, methods and discusses the kernel-based Volterra and the block-based Hammerstein model in depth. Methods and algorithms are investigated through MATLAB simulations and one practical application: a Brüel & Kjær 4010 Mini-Shaker. Problems arising from the implementation of theoretical models to real-world situations are highlighted and tackled.

Keywords: linearization, non-linear system identification, Hammerstein-model, Volterra-model, delayed inverse

1. Bevezetés

Egy lineáris, dinamikus rendszer identifikálásához és irányításához bőséges eszköztár áll rendelkezésre, mivel az elmúlt évtizedekben a lineáris modellalkotás területét alaposan feltérképezték. A legtöbb probléma hatékonyan megoldható az általánosan ismert módszerekkel, melyek általában az impulzusválasz meghatározására vezethetők vissza [1,2].

A valóságban azonban a rendszerek többsége – legalább kis mértékben – nemlineáris. Az alkalmazás megengedheti, hogy torzításukat elhanyagoljuk, és lineáris rendszernek tekintjük őket [3] – bár ekkor egyedül a munkapont kis környezetében kapunk jó eredményt [4]. Egészen más a helyzet, amikor a nemlinearitást figyelembe kell vennünk, mivel azt a precizitás, vagy a nemlinearitás mértéke és egyedi tulajdonsága már megköveteli – ilyen lehet egy telítődéses karakterisztika vagy az a helyzet, amikor a rendszer bemenete és kimenete nem fed át a frekvenciatartományban [5].

Nemlineáris rendszerekkel találkozunk mechanikai gépeknél, szenzoroknál, diódáknál, más elektronikai alkatrészeknél, illetve különféle kapcsolásoknál, nemlineáris ellenállásoknál, rádiófrekvenciás teljesítményerősítőknél, visszhang-mentesítésnél [1,6–8].

Mivel a villamosmérnöki szakma talán az egyik legjobban beágyazott terület, olyan jelenségekre is gondolnunk kell, melyek nem szigorúan ide tartoznak, de megközelítésük, mérésük, és irányításuk igényli egy villamosmérnök jelenlétét. Az általam feldolgozott szakirodalom számos ilyen példát is említ. Nemlineáris rendszerek az izomösszehúzódások és a tüdőben lejátszódó folyamatok, a sejtmembránok potenciáljának küszöbértéke, kémiai folyamatok – mint például a pH-semlegesítés – szabályozása, horkolás modellezése és vizsgálata, illetve a képfeldolgozás [5,9–11].

Nemlineáris rendszerek modellezésére és irányítására azonban nincsen egyetlen jól bevált technika, amit Gasparini [2] így fogalmaz¹ meg: „a nemlineáris szűrők tervezése sokkal inkább művészet, mint száraz tudomány”. Egy-egy optimálisnak tűnő megoldáshoz tapasztalati úton jutunk – ezt támasztja alá a szakirodalom jelentős mennyisége, és a fellelhető eljárások igen nagy száma is.

A dolgozat célja, hogy olyan megoldásokat mutasson be, melyekkel egy gyengén nemlineáris, dinamikus rendszert kompenzálni lehet, vagyis meghatározni annak inverzét és így minimalizálni a be- és kimeneti jel közötti torzítást. A dolgozat gyakorlatorientált abban az értelemben, hogy a modellezést a kompenzálás egy eszközének tekintem, miközben választ keresek olyan kérdésekre is, melyekkel egy-egy tanulmány nem foglalkozik, de a megvalósítás során felmerülhet.

A dolgozat három részre osztható. Az első rész (2.-4. fejezet) az inverz megalkotásával, modellezési technikákkal és nehézségeikkel foglalkozik. Részletesen bemutatom a Volterra- és a Hammerstein-modellt. Fontosnak tartottam, hogy a dolgozat egységet képezzen úgy, hogy a feladat megvalósításához minden szükséges eszközt tartalmaz – ezért a második részben (5. fejezet) ezekre térek ki. Bemutatom a leggyakrabban előforduló nemlineáris karakterisztikákat, a nemlinearitás lehetséges mérőszámait, algoritmusokat, inverzmeghatározási technikákat és azt,

¹ [2], 772. oldal

hogyan lehet modellek paramétereit illeszteni. A harmadik részben (6.-7. fejezet) a Hammerstein- és Volterra-modell működését vizsgálom szimuláció és egy valós gyakorlati alkalmazás – egy shaker – segítségével. Az eredmény, amit itt elérek kiterjeszhető más, hasonló eszközökre is, ugyanis ugyanis egy olyan struktúra, mely adaptívan kalibrál egy szenzort és így kiküszöböli az öregedést, megnyitja az utat az érzékeny műszerek tömegtermelése felé [12].

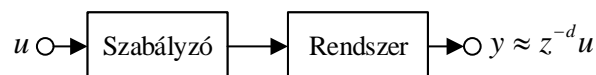
2. Nemlineáris rendszerek modellezése

Egy mérő-, vagy szabályozó rendszer elemeitől lineáris viselkedést várunk el, mivel a lineáris eszközök kezelése könnyebb, és működésük hatékonyabb, mint nemlineáris társaiké. Példaként különféle jelátalakítókat (szenzorokat), rádiófrekvenciás teljesítményerősítőket és a széles sávú kommunikációs rendszereket érdemes megemlíteni [13], ahol a működési tartományt nagyban befolyásolja, hol tekinthető lineárisnak az eszköz.

A gyakorlati életben a nemlineáris rendszereket ezért gyakran kompenzálni – linearizálni – kell. A cél ekkor az, hogy a rendszer és a vele sorosan kapcsolt szabályzó együttes átviteli függvénye az egységnyi erősítésnek feleljen meg. A kontrollert leggyakrabban valamilyen digitális jelfeldolgozó eszköz: PC, FPGA, vagy DSP kártya.

2.1. Az inverz jelentősége és meghatározásának problémái

Kétféle elrendezés közül választhatunk aszerint, hogy a szabályzó megelőzi-e a szakaszt vagy sem. Ha a szakasz van előbb, nem tudunk fizikai kimenettel – mint a gyorsulás, vagy hang – rendelkező mechanikai rendszereket kompenzálni, így dolgozatomban a másik lehetőséget vizsgálom. Ezt az elrendezést mutatja az 1. ábra, ahol u jelöli a bemenetet és y a kimenetet. A két jel – esetleg a kimenet és a $z^{-d}u$ késleltetett bemenet – közötti torzítás akkor a legkisebb, ha a szabályzó megegyezik a szakasz inverzével [14].



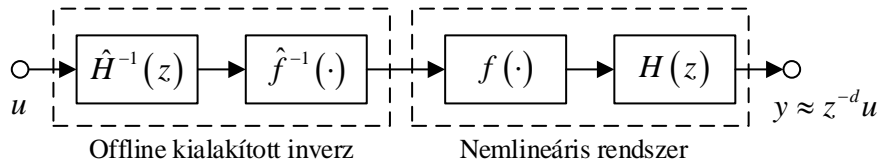
1. ábra. Szakasz és szabályzó kompenzáláshoz

Widrow [14] felhívja arra a figyelmet, hogy a nemlineáris szakaszoknak szigorúan véve nem létezik inverze, de a lineáris szűrőkhöz kifejlesztett eljárások általában nemlineáris szűrőknél is működnek. Ezt saját szimulációs kísérleteim is alátámasztják. Ezért a továbbiakban a két témakört egyként kezelem.

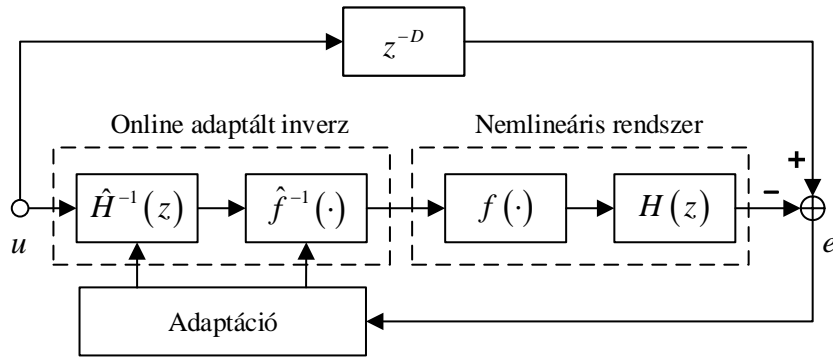
Az inverz előállításakor három jelentős probléma merül fel:

- Diszkrét időben, minimálfázisú szakasz esetén — vagyis amikor az összes zérus az egységkörön belül helyezkedik el - az inverz összes pólusa az egységkörön belülre kerül. Ha azonban egy diszkrét idejű átviteli függvény zérusai az egységkörön kívül helyezkednek el, az inverz pólusai az egységkörön kívülre esnek, ami azt jelenti, hogy az inverz nem stabil [14]. A problémát úgy oldhatjuk meg, hogy az inverz meghatározása helyett a késleltetett inverzet identificaljuk. A késleltetett inverz definíciója az (1) szerinti [15]:

$$H(z)H_D^{-1}(z) \simeq z^{-D} \quad (1)$$



2. ábra. Nemlineáris rendszer kompenzálása offline meghatározott inverzzel



3. ábra. Nemlineáris rendszer kompenzálása online meghatározott inverzzel

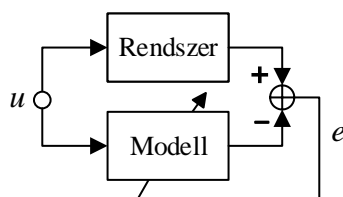
- Instabilitást okozhat az adaptációs hurok amennyiben abban dinamika van [15]. A visszacsatolásra és a helyes fázisra ezért különösen ügyelni kell, csak speciális esetben lehet közvetlenül visszacsatolni a hibajelel.
- Az adaptálandó struktúra komplex megközelítést igényelhet egyes blokkalapú modelleknél. Ebben az esetben nem elegendő egyetlen hibajel, hanem azt esetenként elemeire kell bontani, és a struktúra egyes részeit külön-külön ezekkel tanítani. Ezzel a kérdéssel később, a Hammerstein-modell tárgyalása után fogok foglalkozni.

2.2. Az inverz online és offline meghatározása

Az inverz előállítására kétféle megközelítéssel lehetséges: online és offline identifikációval.

Offline identifikációnál a kompenzáló rendszer modelljéből indulunk ki, melyet adaptív vagy analitikus eszközökkel invertálunk. Az eljárás szemléltetésében segít a 2. ábra. Tegyük fel, hogy egy rendszer két sorba kapcsolt blokkal modellezhető. Az egyik blokk egy memóriamentes nemlinearitás, a másik pedig egy lineáris dinamikus szűrő. Ha ezeket külön-külön identifikáljuk és invertáljuk, majd a rendszer elé kötjük őket, képesek vagyunk a nemlineáris viselkedést kompenzálni. Az invertálás technikájára később térek majd ki.

A szakasz megváltozásakor a teljes műveletet újra el kell végeznünk, miközben általában alapvető követelmény egy gyakorlati alkalmazással szemben, hogy képes legyen a lassan válto-



4. ábra. Rendszer identifikálásának struktúrája

zó környezethez folytonosan alkalmazkodni. Az inverz meghatározása ekkor valamilyen online, adaptív megközelítést igényel.

Az inverz online meghatározásának alapsémáját a 3. ábra szemlélteti. A szabályzó ekkor egy nemlineáris, dinamikus szűrő, melynek kimeneti jelét a rendszerre vezetjük. A késleltetett bemeneti jellel hasonlítjuk össze a rendszer kimenetét, és az így kapott hibát használjuk fel az adaptáció során.

A 3. ábrán látható struktúra azonban közvetlenül nem használható fel a kompenzáláshoz, ugyanis az adaptációs hurokbeli visszacsatolás instabilitást okozhat [15]. Erre nyújt megoldást az ELMS és az XLMS algoritmus, melyeket később mutatok majd be.

Az adaptív kompenzáló struktúrák felhasználhatják a rendszer modelljét, vagy az inverz egyszerűsített modelljét is.

2.3. Nemlineáris rendszerek modelljei és jellemzőik

Az inverz megalkotásán túl számos más célunk is lehet, amikor rendszert modellezünk. Egy jó modell előrejelzi a rendszer viselkedését, magyarázatot ad bizonyos jelenségekre, és megkönnyíti a hibakeresést is. Segít a rendszer vizsgálatában és optimalizációjában, ha a közvetlen kísérletezés túlzottan hosszú időt vagy túl sok erőforrást venne igénybe [16, 17].

Modellalkotáskor olyan szabályokat keresünk, melyek jól leírják egy szakasz viselkedését [18]. A cél az, hogy azonos gerjesztésnél a rendszer és modell válasza közel megegyezzen. A 4. ábra ezt szemlélteti: ha u a gerjesztés, a cél a rendszer és a modell kimenete közötti eltérés, az e hibajel nullához közelítése.

2.3.1. Nemlineáris rendszerek definíciója és modellezésének lépései

A nemlineáris rendszereket Ljung [19] definíciója szerint határozhatjuk meg: legyen egy rendszer bemenete u és kimenete y . A modell ekkor a közöttük lévő olyan $f(\cdot)$ kapcsolat, amely az aktuális kimenetet a korábbi be-, és kimeneti értékekhez rendeli. A rendszer nemlineáris, ha az $f(\cdot)$ függvény nemlineáris. A lineáris rendszerekkel ellentétben a nemlineáris rendszerekre nem érvényes a szuperpozíció elve [16].

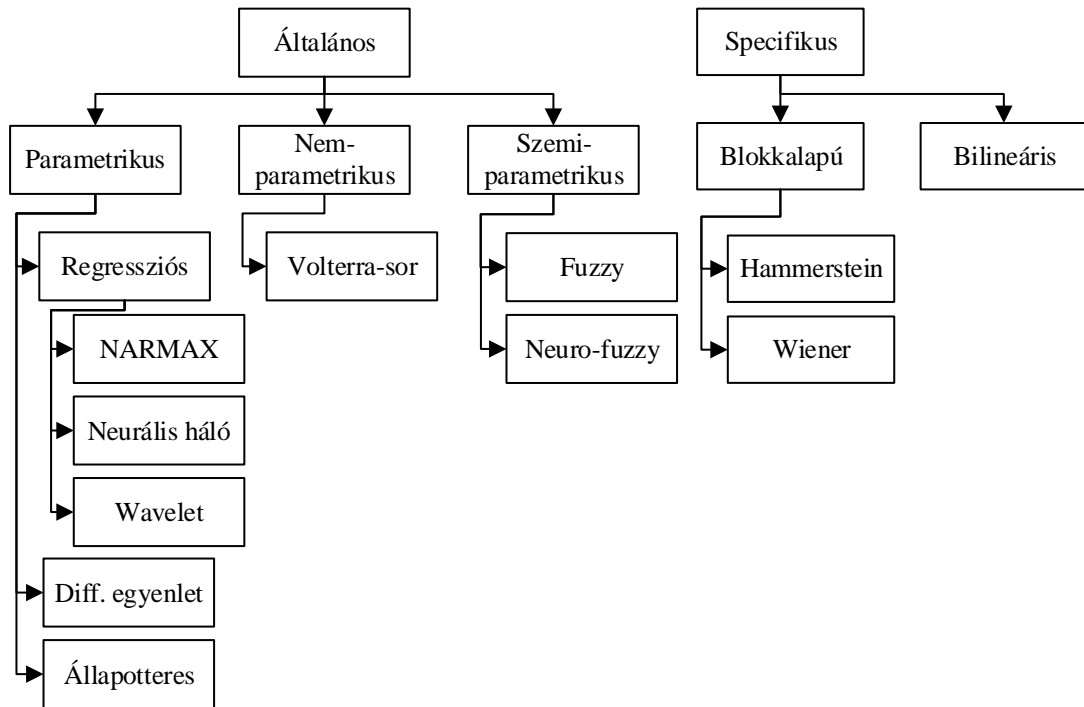
Modellezésénél észben kell tartanunk, hogy egy nem megfelelően választott nemlineáris modell vagy adaptív struktúra rosszabb eredményre vezethet, mintha egy lineáris modellel közelítenénk

meg a problémát [17]. Azért, hogy ezt elkerüljük, Nelles [17] nyolc olyan lépést gyűjtött össze, melyek egy jó nemlineáris dinamikus modell identifikálásához szükségesek:

1. A modell bemeneteinek kiválasztása: a dolgozatban csak egy bemenetű – egy kimenetű rendszerekkel foglalkozom, de előfordulhat, hogy egy rendszernek több bemenete és több kimenete van – kémiai, biológiai folyamatok esetén ez a leggyakoribb helyzet. Ekkor lehetőség van arra, hogy minden bemenetet felhasználjunk vagy kipróbáljuk az inputok összes lehetséges kombinációját és ezekből választunk. Főkomponens-elemzéssel és korreláció számításával is felderíthetjük a kapcsolat szorosságát a bemenetek és a kimenet között.
2. A vizsgálójel megválasztása. Akkor tudunk jól választani, ha előzetes információval rendelkezünk a rendszerről. Gyakran alkalmaznak véletlen zajt vagy szinuszos jelet gerjesztésnek. (Utóbbira is érdemes egy kisebb amplitúdójú véletlen zajt szuperponálni, hogy a frekvenciafüggő tulajdonságok jobban mérhetőek legyenek [14].)
3. A modell típusának meghatározása: el kell döntetünk, hogy online vagy offline végezzük az identifikációt, illetve, hogy az adott modell mennyire alkalmas optimalizációra, szimulációra, irányítási feladatokra. Azt is mérlegelni kell, hogy a megrendelő mennyire bízik egy technikában – például a neurális hálók esetén.
4. Dinamikus viselkedés modellezése. A rendszermodellek – mint amilyenek a később bemutatott blokkalapú modellek is – legtöbbször egy-egy lineáris szűrő köré épülnek. Ez modellezi a rendszer dinamikus tulajdonságait. A struktúra stabilitásukat befolyásolja a szűrő megválasztása. Egy FIR szűrő mindig stabil, míg ez egy IIR szűrőről ez nem mondható el.
5. Modell fokszámának megválasztása: ezt általában próbálgatással lehet jól meghatározni, de korlátot jelent a számítási kapacitás, illetve a késleltetés is. Egy FIR szűrő esetén, ha túl kevés együttthatót használunk, bizonyos mértékű időbeli késleltetést már nem tudunk modellezni.
6. Algoritmusok és struktúrájuk: befolyásolják a modell stabilitását és a konvergencia sebességét is, illetve figyelembe kell vennünk a számítási kapacitást is.
7. A modell paramétereinek megválasztása.
8. Validálás. A modellezési folyamat zárásaként azt kell megvizsgálni, hogy a modell valóban jól írja-e le a rendszer viselkedését – ezzel igazoljuk, hogy az eddigi lépések helyesek. Fontos, hogy olyan adatokon végezzük a tesztet, melyeket nem használtuk fel a modell kialakítása során. A témát részletesen körüljárja Billings [20].

2.3.2. Nemlineáris modellek csoportosítása

Számos olyan technika született, mellyel nemlineáris rendszereket lehet modellezni. Röviden bemutatom az általam legfontosabbnak ítélteteket. Többféle csoportosítás is létezik, ezek közül



5. ábra. Nemlineáris modellek csoportosítása Unbehauen alapján. [16] alapján.

kettőt érintek, és rövid magyarázatot fűzök az egyes pontokhoz. A modellekről áttekintést nyújt az 5. ábra.

Parametrikus – nemparemetrikus

A nemlineáris modellek egy részletes csoportosítását adja [16], aki két csoportra osztja a nemlineáris modelleket: általánosakra (parametrikus, nemparemetrikus, szemi-parametrikus) és specifikusakra (blokkalapú, bilineáris modell).

1. Parametrikus modellek: egy parametrikus modell véges számú paraméterrel rendelkezik. Ljung [19] precíz matematikai definíciót is megad.

- Regresszió alapuló modellek
 - NARMAX (*nonlinear autoregressive moving average with exogenous inputs*): egy általános modellkeret, mely a (2) egyenleten alapul.

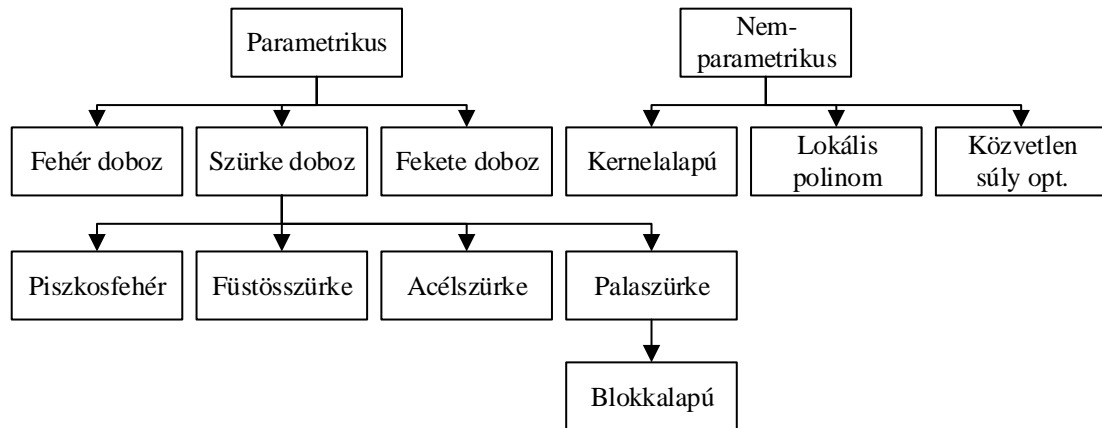
$$y_k = f(y_{k-1}, y_{k-2}, \dots, u_{k-d}, u_{k-d-1}, \dots, e_{k-1}, e_{k-2}, \dots) + e_k \quad (2)$$

A kimenetet y , a bemenetet u , a zajt e , a késleltetést d , illetve a nemlineáris függvényt $f(\cdot)$ jelöli. Minden tag véges számban szerepel az egyenletben [17]. Alkalmazására példa lehet egy egyenáramú motor modellezése [21].

- Neurális hálók: olyan struktúrák, melyek megpróbálják a biológiai neuronok működését utánozni. A témát részletesen bemutatja [17]. Megjelenhetnek önálló eszközként, de egy másik technikát is kiegészíthetnek [14].
 - Wavelet-ek: egy olyan módszer, mely a wavelet-dekompozíciót használja fel arra, hogy nemlineáris modelleket alkosson. Billings és Wei [22] részletesen bemutatja az eljárást.
- Differenciálegyenleten alapuló modellek. Ezt a modelleszaládot részletesen bemutatja [20]. Differenciálegyenletek olyan jelenségeket is leírhatnak, mint a hiszterézis, ezért blokkalapú modellek építőelemei is lehetnek .
 - Állapotterez modellek: a modelleszalád két egyenlettel írja le a nemlineáris rendszert: $x_{t+1} = f_t(x_t, u_t, v_t, \theta)$ és $y_t = h_t(x_t, u_t, e_t, \theta)$. x az állapotvektort, y a kimenetet, u a bemenetet jelöli, θ az $f(\cdot)$ és a $h(\cdot)$ nemlineáris függvény paramétereinek vektora, illetve v és e normális eloszlású fehér zaj. A témát részletesen körüljárja Schön et. al. [23] és Nelles [17]).
2. Nemparametrikus modellek: a nemparametrikus modellek alapvető feltevése, hogy az az adathalmaz, amire illesztjük a modellt, nem írható le végesen sok paraméterrel – így például előfordulhat, hogy az adatok számának növelésével a paraméterek száma is nő. Ljung [19] precízen definiálja ezt. Nemparametrikus modell például a Volterra-modell, amely a bemeneti értékek késleltetett értékeinek és azok keresztszorzatainak lineáris kombinációjával becsüli a rendszer kimenetét [17].
 3. Szemiparametrikus modellek: ide a fentiek keveréke tartozik, például a fuzzy-algebrán alapuló fuzzy-modellek, és azok neurális hálóval való kiegészítése, a neuro-fuzzy modellek.
 4. Blokkalapú modellek: memóriamentes nemlinearitások és lineáris dinamikus blokkok párhuzamos és sorba kapcsolásával nyerjük a modellt. Többek között ide tartozik a Hammerstein és a Wiener-modell, illetve ezek különféle variációi. Ezekről később részletesebben is írok.
 5. Bilineáris modell: részletesen bemutatja Diniz [24]. A kimenetet a kimenet korábbi értékeivel, a bemenet aktuális és korábbi értékeivel, illetve a be- és kimenet kombinációinak szorzataival becsüli.

Fehér doboz, szürke doboz, fekete doboz

A modellalkotásnak két kétféle megközelítése létezik attól függően, hogy milyen előzetes információval rendelkezünk a rendszerről. Ha ismerjük a rendszer pontos belső fizikai felépítését és működését, matematikai egyenletekkel leírhatjuk, hogyan viselkedik. Ekkor fehér doboz modelltől beszélünk. A természetben előforduló folyamatok azonban sokszor bonyolultabbak annál, hogy



6. ábra. Nemlineáris modellek csoportosítása Ljung [19] alapján.

fizikai törvényekkel és egyenletekkel ragadjuk meg őket. Ha csak a be-, és kimenet mérésével és az adatokra illesztéssel lehetséges a modellalkotás, fekete doboz modellről beszélünk [16].

Sjöberg et. al. [18] a fehér és fekete dobozon kívül még egy harmadik lehetőséget is megemlít: a szürke doboz modelleket. Ezt a hármas tagolást Ljung [19] bővíti tovább.

Ljung [19] a fehér-fekete doboz modelleket a parametrikus modellek változataiként írja le, míg szerinte a nemparametrikus modellek közé tartoznak a kernelalapú, a lokális polinom alapú, közvetlen súlyoptimalizációs modellek. Az ilyenfajta csoportosítás áttekintését mutatja be a 6. ábra.

- Parametrikus modellek

- Fehér doboz modellek (*white box models*): A nemlineáris rendszert leíró fizikai modellt pontosan ismerjük az összes paraméter értékével együtt [18]. A rendszer lényegében előre felírható algebrai egyenletek összessége [19]. A rendszer jó megértésére szolgál, magasan megbízható, de sok időt vehet igénybe, főleg csak egyszerű esetekben alkalmazható [17].
- Szürke doboz modellek (*grey box model*): Valamennyire ismerjük a fizikailag lezajló folyamatokat, vagy feltételezzük azokat, de számos paraméter ismeretlen [18].
 - * Piszkosfehér modellek (*off-white models*): Olyan fehér doboz modellek, ahol néhány paraméter értéke ismeretlen vagy bizonytalan [19].
 - * Füstösszürke modell (*smoke-grey models*): Ekkor az a feladat, hogy olyan nemlineáris transzformációt találjunk, mely után a rendszert lineáris egyenletekkel írhatjuk le. Ljung [19] megfogalmazása szerint a művelet nem tarthat tovább 10 percnél.

- * Acélszürke modellek (*steel-grey models*): Ebben az esetben nem elegendő egyetlen fizikai modell arra, hogy a rendszert jól leírjuk. A különböző munkapontok környékére, vagy különböző környezeti feltételekre más-más lineáris modellt kell alkotni [19].
- * Palaszürke modellek (*slate-grey models*): valamennyi rálátásunk van arra, hogy milyen változókat kell használnunk [19]. A szerzők szerint ide tartoznak a blokk-alapú modellek.
- Fekete doboz modell (*black box model*): Ebben az esetben vagy a belső fizikai felépítést nem ismerjük, vagy nem használjuk fel ezt a tudást. Olyan modellstruktúrát választunk, ami a múltban sikeresen bizonyított [18]. Be és kimeneti adatok jelentik az egyetlen információforrást, de gyorsan fejleszthető, ugyancsak segítheti a rendszer folyamatainak megértését. Az adathalmaz azonban korlátozza a pontosságot [17].

- Nemparametrikus modellek

- Kernelalapú módszerek (*kernel methods*):
- Lokális polinom alapú módszerek (*local polynomial methods*)
- Közvetlen súly optimalizáció (*direct weight optimization*)

Mivel nemlineáris rendszereknél nem lehet a be- és kimeneti jelek közötti kapcsolatot egyetlen univerzális eszközzel megragadni [18] és ezen túlmenően, leggyakrabban sem a részletekre kiterjedő pontos belső felépítést, sem a ki- és bemenet közötti matematikai kapcsolatot nem ismerjük, általában fekete doboz modellek felől kell megközelíteni egy-egy problémát [4].

2.3.3. Hogyan válasszunk a modellek közül?

Hogyan válasszuk ki a legjobb modellt, ha több is rendelkezésünkre áll? A modellek nem csak típusban (Volterra-, Hammerstein-modell), hanem egy-egy típuson belül paramétereikben, és a paraméterek számában is eltérhetnek egymástól. Érdeemes a lehető legkevesebb paraméterrel megoldani a feladatot, és Nelles [17] felhívja arra is a figyelmet, hogy olyan modellező technikát érdemes keresni, ami speciális esetenként tartalmazza a teljesen lineáris modellt is.

Célszerű a modell teljesítményéhez egy számot is rendelni, amivel a modellszelekció részben automatizálhatóvá válik. A legegyszerűbb megoldás az, ha elegendő lépésszám – a szűrő beállása – után az átlagos abszolút hiba alapján döntünk. Ennek a technikának a hátránya, hogy a modell túltanulhat, ami ahhoz vezethet, hogy ismeretlen adatokon vagy zajos környezetben nem működik jól.

Ezért olyan mértéket kell választani, ami a pontosság mellett az „általánosságot is méri” [25].

Egy ilyen lehetőség az FPE (*final prediction error*) kiszámítása [25]. A (3) képletben $\hat{\theta}$ jelöli a becsült paramétereket, N a be- és kimeneti adatpontok számát, m pedig a modell paramétereinek számát.

$$FPE = \frac{SSE(\hat{\theta})}{N} \left(\frac{N+M}{N-M} \right) \quad (3)$$

$$SSE(\hat{\theta}) = \sum_{n=1}^N (d(n) - y(n))^2 \quad (4)$$

Egy másik módszer, az NMSE (*normalized mean squared error*) akkor vezet optimális modellhez, ha a bemenet normális eloszlású véletlen zaj [9]. A (5) képletben d a rendszer kimenetét, \hat{y} a szűrő kimenetét jelöli.

$$NMSE = \left(\frac{\text{var}(d - \hat{y})}{\text{var}(d)} \right) \quad (5)$$

3. A Volterra-modell

A Volterra-szűrő (*VAF, Volterra adaptive filter*) az egyik leggyakoribb fekete doboz modell. Bár a Volterra-sor végtelen tagból áll, a gyakorlatban véges számú tagot használnak és a sort a (6) formában írják fel [5]. Ezt a modellt nemlineáris véges impulzusválaszú (*non-linear finite impulse response, NFIR*) modellnek is nevezik [26].

$$\begin{aligned}
 y(n) = h_0 + & \sum_{m_1=0}^{N-1} h_1(m_1) u(n-m_1) + \\
 & \sum_{m_1=0}^{N-1} \sum_{m_2=0}^{N-1} h_2(m_1, m_2) u(n-m_1) u(n-m_2) + \\
 & \sum_{m_1=0}^{N-1} \sum_{m_2=0}^{N-1} \dots \sum_{m_p=0}^{N-1} h_p(m_1, m_2, \dots, m_p) u(n-m_1) u(n-m_2) \dots u(n-m_p)
 \end{aligned} \tag{6}$$

A sorban szereplő h_0 konstansról általában azt tesszük fel, hogy értéke 0. A Volterra-modell előnyei:

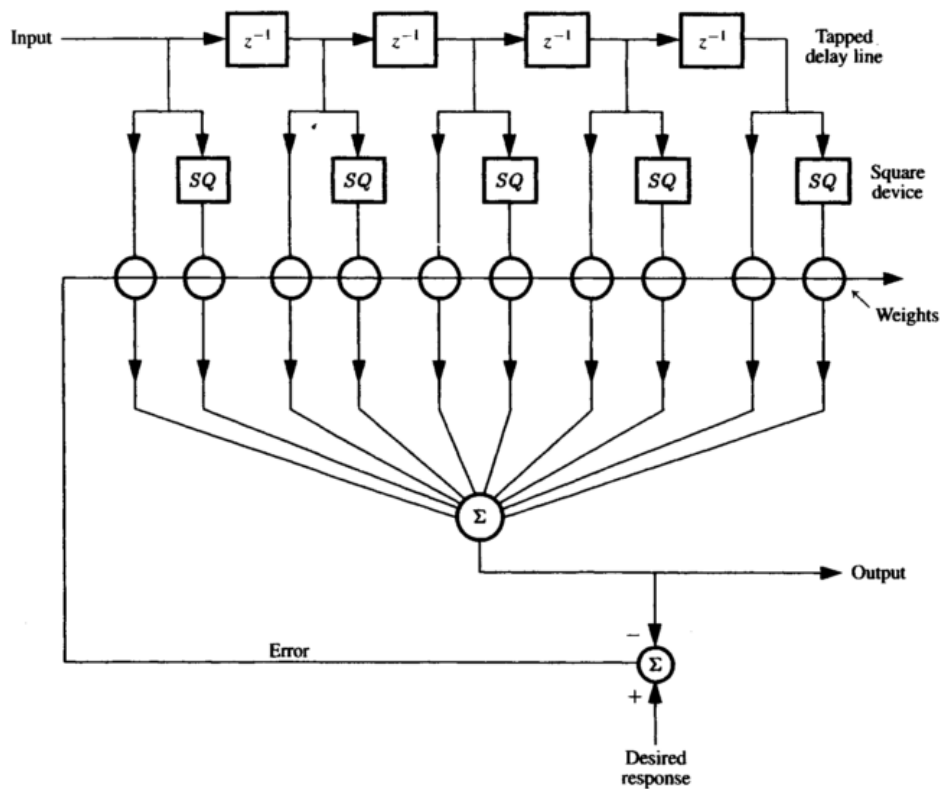
- Egyszerű és könnyen implementálható. Egy lineáris szűrőt nagyon könnyű átalakítani úgy, hogy a Volterra-modellnek megfelelően működjön.
- Ha a nemlinearitás könnyen Taylor-sorba fejthető, a Volterra-szűrő gyorsan konvergál.
- Nemlineáris dinamikát is modellez az egyszerűbb blokkalapú modellekkel szemben. Ezzel egy nagyon sokoldalú eszközt ad a szakemberek kezébe.

Hátrányai között említhető, hogy:

- Nem tudja jól modellezni a szakadó függvényeket [5].
- A sok szabad paraméter miatt általában csak gyenge nemlinearitásoknál alkalmazzák [4]. Erős nemlineáris függvények ugyanis magas hatványokat igényelnek, ami a számítási igény növekedéséhez, illetve oszcilláló viselkedéshez vezet. Ez instabillá teheti a szűrőt [17].
- Nemcsak erős nemlinearitásoknál nő meg (exponenciálisan) a számítási igény, hanem nagyobb sáv szélességnél is – például rádiófrekvenciás teljesítményerősítőknél [7].

A számítási igényt úgy csökkenthetjük, hogy rekurzívva tesszük a (6) formulát [5] vagy elhagyunk bizonyos tagokat [2]. Utóbbi esetben például a bemeneti értékek hatványai szerepelhetnek csak a sorban (7). A dolgozat későbbi részében ezt a formát használom majd fel. Widrow [14] szemléletes ábrája segítheti az áttekintést (7. ábra).

$$y(n) = \sum_{i=1}^I \sum_{m=0}^{N-1} h_{i,m} u^i(n-m) \tag{7}$$



7. ábra. Egyszerűsített Volterra-modell blokkvázlata.

Az ábra forrása: [14], 305. oldal.

Akkor tanácsos Volterra-modellt használni, ha a modellezendő rendszer statikus karakteristikája megközelítőleg Taylor-sorba fejthető és a sor első néhány hatványa már jól leírhatja a rendszert.

Ekkor is figyelni kell azonban egy korlátozó jelenségre. Ha a gerjesztés felső sávhatára f_u^{max} , akkor második hatványáé $2f_u^{max}$, a harmadiké $3f_u^{max}$, és így tovább, az n . hatványáé nf_u^{max} . Diszkrét idejű jelfeldolgozásnál, ha a jel felső sávhatára meghaladja a mintavételi frekvencia felét, átlapolódás lép fel. Adaptáció során az átlapolódó komponensek a nemlineáris Volterra-szűrő rendellenes működéséhez vezethetnek. Ezt a dolgot későbbi részében szimulációval is alátámasztom.

4. Blokkalapú modellek és a Hammerstein-modell

Egy dinamikus rendszer viselkedése jól közelíthető blokkok, vagyis statikus nemlinearitások és lineáris dinamikus elemek – lineáris dinamikus szűrők – sorba kapcsolásával [1] vagy általánosabb esetben soros és párhuzamos kapcsolásával [27] is.

A blokkalapú modelleknek számos előnyös tulajdonsággal rendelkeznek:

- Alacsonyabb számítási kapacitást igényelnek, mint a Volterra-sor [1].
- Ha a Volterra-modell véges memóriával rendelkezik, leképezhető véges számú, párhuzamosan kapcsolt LN² vagy LNL³ blokkra [6].
- A szakirodalomban széles körben elterjedtek, így számos specifikus alkalmazási területen vizsgálták már őket, például elektromechanikai rendszerek, rádiófrekvenciás alkalmazások, beszédfeldolgozás, és kémiai folyamatok irányításánál is [27].
- Jól megválasztott nemlineáris blokkal szakadó, nem invertálható, vagy más erős nemlinearitásokat is jól modelleznek.

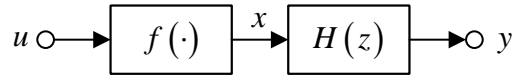
Hátrányuk:

- Egyes folyamatok csak bonyolult struktúrákkal közelíthetőek, melyek adaptálása körülményes folyamat. Az egyszerű blokkalapú modellek a Volterra-sorral szemben nem képesek a nemlineáris dinamikát modellezni, így hajlamosak túlzottan leegyszerűsíteni egy rendszer működését.
- Az online és offline identifikáció során a modell kimenetét hasonlítjuk a rendszer kimenetéhez, majd az eltérést alapul véve módosítjuk a modellt addig, amíg a rendszer és a modell válasza azonos nem lesz. Az egyetlen rendelkezésre álló hibajel megnehezíti az egyes blokkok tulajdonságainak meghatározását. Ha csak egy dinamikus lineáris, és egy statikus nemlineáris blokkot kötünk sorba, máris szemben találjuk magunkat azzal a problémával, hogy melyik blokk tartalmazzon és mekkora erősítést. Ha két dinamikus lineáris blokkunk van, hova tegyük a késleltetést? Így egy blokk alapú modell adaptálása számos problémát felvet. Az egyes blokkok eltérő hibajelét igényelhetnek, de csak egy áll elő természetes módon a kimenetnél. Ezzel a kérdéssel később fogok foglalkozni.

Azért, hogy érzékeltessem, milyen sokszínű a blokkalapú modellek családjá, bemutatok néhány változatot, majd arról írok, hogyan lehet megvalósítani az egyes blokkokat.

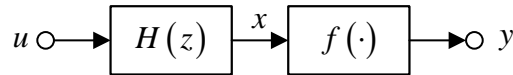
²Lineáris dinamikus - statikus nemlineáris.

³Lineáris dinamikus - statikus nemlineáris - lineáris dinamikus.



8. ábra. A Hammerstein-modell blokkvázlata.

A memóriamentes nemlinearitás megelőzi a lineáris dinamikus blokkot.



9. ábra. A lineáris dinamikus szűrő megelőzi a statikus nemlinearitást.

4.1. Néhány lehetőség

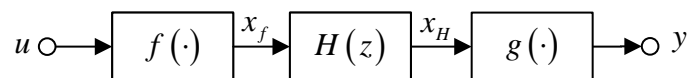
A blokkalapú modellek közül a legegyszerűbb a Wiener-, és a Hammerstein-modell, mivel csak két-két blokkból állnak. A Hammerstein-modellben az u bemenet előbb az $f(\cdot)$ memóriamentes nemlinearitáson, majd a $H(z)$ lineáris dinamikus szűrőn halad keresztül (8. ábra). A Wiener-modellben a sorrend pont fordított (9. ábra).

A két modell egyszerűsége ellenére számos alkalmazási területen bizonyított, mivel igen rugalmasak [9]. Wiener-, vagy Hammerstein-moddellel például kezelni lehet hiszterézist [28] és pH-szabályzást [29], illetve rádiófrekvenciás teljesítményerősítők esetén is sikerrel alkalmazták őket⁴ [7].

A két blokkból természetesen bonyolultabb struktúrákat is építhetünk. A lehetőségeknek elsősorban az szab határt, hogy a blokkok számának növelésével a paraméterek meghatározása egyre nehezebbé válik [7]. A Wiener-, és Hammerstein-modell egyszerű továbbfejlesztésére mutat példát a 10. ábra, ahol a lineáris szűrőt két nemlineáris függvény fogja közre.

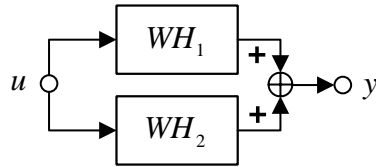
A blokkalapú modellek Hammerstein-, és Wiener-moделlekből is felépülhetnek úgy, hogy azokat párhuzamosan vagy sorosan összekapcsoljuk egymással [6], vagy visszacsatolhatjuk adott blokkok kimenő jelét, ahogyan azt [7] is teszi. Előbbire a 11. ábrán, utóbbira a 12. ábrán láthatunk egy-egy példát. Visszacsatolást ott érdemes alkalmazni, ahol a folyamat megköveteli, hogy a modell hosszú memóriával rendelkezzen [7].

A Wiener-, és Hammerstein-modell egyszerű változatát úgy is továbbfejleszthetjük, hogy nem

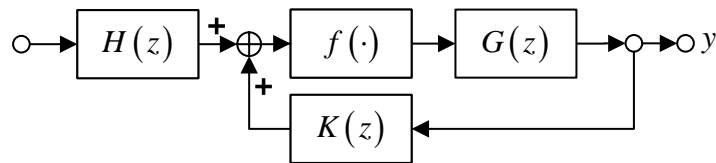


10. ábra. A Wiener-Hammerstein-modell. [27] alapján.

⁴Nemlineáris teljesítményerősítők esetén érdemes visszacsatolni a kimenő jelet.



11. ábra. Párhuzamosan kapcsolt Wiener-Hammerstein-modellek. [6] alapján.



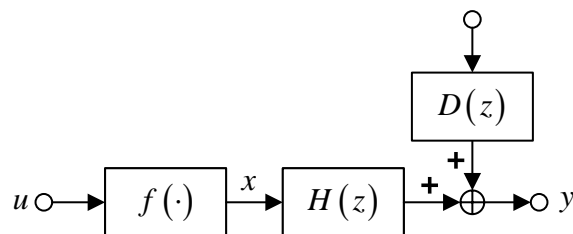
12. ábra. Visszacsatolt blokkalapú modell. [7] alapján.

újabb blokkot vagy visszacsatolást adunk a struktúrához, hanem figyelembe veszünk zajforrásokat is. Gyakran feltételezik, hogy a mérést zaj terheli és ezért szűrt zajnak adnak a rendszer kimenetéhez [30], [1]. Ezt mutatja a 13. ábra.

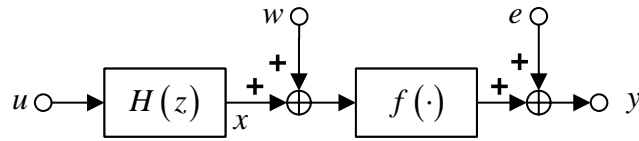
Hagenblad és Ljung [29] egyik modellje a mérési zajon túlmenően rendszerzajt is figyelembe vesz (14. ábra).

4.2. A Hammerstein-modellről részletesen

A továbbiakban a Hammerstein-moddellel fogok foglalkozni. Azért a Hammerstein-modellt választottam a Wiener-modell helyett, mert linearizálási feladatokra alkalmasabb – elegendő a nemlineáris függvényt invertálni és az irányítandó szakasz elé kötni. (Sajnos ezért a tulajdonságért cserébe identifikálni is nehezebb.) A következőkben körüljáróm, milyen eszközökkel lehet megvalósítani az egyes blokkokat.



13. ábra. A Hammerstein-modell szűrt zajjal. [30] és [1] alapján.



14. ábra. Wiener-modell folyamatot és mérést terhelő zajjal. [29] alapján.

4.2.1. A memóriamentes nemlinearitás modellezése

A Hammerstein-modell egyik blokkja egy olyan paramétereztető függvény, mellyel a nemlinearitást ragadjuk meg. Azért kell paramétereztetőnek lennie, hogy adaptív algoritmussal állíthassuk az alakját.

A nemlinearitásokat két csoportba lehet sorolni [18]. Az egyik csoportra az jellemző, hogy bázisfüggvényei lokálisak, vagyis csak egy-egy részintervallumon értelmezettek, míg a másik csoport bázisfüggvényei globálisak. Előbbi csoportba soroljuk a spline függvényeket, utóbbiba a polinomokat. Sjöberg [18] ezt részletesen bemutatja, és általános tanácsokat arról, hogyan kell nemlineáris függvényt választani:

- Meg kell állapítani, mennyire lineáris a rendszer. A kimenet azonos nagyságban és irányban változik-e a bemenettel, illetve érdemes megbecsülni az időállandókat is.
- A lehető legegyszerűbb modellre kell törekedni. A nem lényeges paraméterek csak nagy hibával becsülhetőek, és ezért jelentősen növelhetik a maradó hibát.
- Sok esetben egy hozzávetőleges fizikai modell adhat ötleteket, milyen nemlinearitásra számíthatunk.
- A modellt valós adatokon is tesztelni kell.

A MATLAB [31] rendszer identifikációs toolbox-ában a következő lehetőségek közül választhatunk: szakaszonként lineáris függvény, neurális háló, wavelet-hálózat, telítődéses karakterisztika, holsáv és polinom. A szakirodalom ezen túlmenően leggyakrabban a következőket alkalmazza:

- Szakaszonként lineáris függvények. A módszer azon alapul, hogy bármely nemlineáris karakterisztika közelíthető szakaszonként lineáris függvényekkel. (Szenzor adatlapoknál is gyakran alkalmazzák.) Ez a megközelítés megkönnyíti az inverz számítását, de ügyelni kell az osztópontok helyes megválasztására [25].
- Szakadó lineáris függvények. A szakaszonként nemlineáris függvények egy egyszerűsített változata, amikor feltételezzük, hogy a nemlineáris karakterisztika az x-tengely pozitív és negatív felén is lineáris, csak 0-ban törik meg, vagy szakad [1].

- Polinomok (8). Ilyen módszert használ [6], [10] és [32] is. Nagy előnye, hogy az együtthatók és egy jól összeállított bemenetiérték-vektor keresztszorzataként a teljes Hammerstein-modell kimenetét egy lépésben megkapjuk.

$$x(n) = \sum_{m=0}^M p_m x^m(n) \quad (8)$$

A polinomokkal kapcsolatban azonban sok kritika felmerült:

- Erős nemlinearitások esetén pontatlan, főleg az intervallum széleinél [2].
 - A rendszer hajlamos oszcillációra [2].
 - Egyetlen paraméter megváltoztatásával az egész nemlineáris függvény alakja jelentősen megváltozhat [2].
 - Egy paraméter állítása nem lokális, hanem az egész függvényre kihat [9].
 - Nehezen modellezi, ha a karakterisztikában törés vagy szakadás van [9].
- Szakadó polinomok (9). Vörös [33] úgy oldotta meg a polinomalapú megközelítés egyik problémáját, hogy két részre vágta az x-tengelyt, és külön-külön illesztett egy-egy polinomot rájuk.

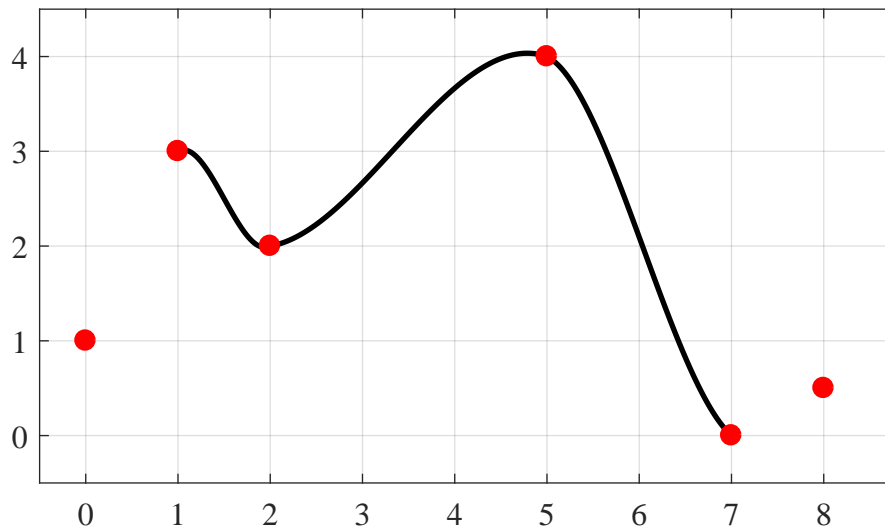
$$y(t) = \begin{cases} f(x(t)) = \sum_{k=0}^K f_k x^k(t), & \text{ha } x(t) \geq 0 \\ g(x(t)) = \sum_{k=0}^K g_k x^k(t), & \text{ha } x(t) < 0 \end{cases} \quad (9)$$

4.2.2. Spline-függvények

A spline függvények olyan függvények, melyeket szakaszonként polinomokkal írunk le. A szakaszok úgy csatlakoznak egymáshoz, hogy a szakasz határán a függvény folytonos legyen, és a deriváltak is megegyezzenek egymással. A függvények értelmezési tartományát kontrollpontok határolják, amik meghatározzák egy-egy intervallum viselkedését. Spline-függvényeket széleskörűen alkalmaznak számítógépes grafika területén, neurális hálóknál, mivel rugalmasságuk nagyban javítja a teljesítményt [34], [35] és [36]. Robotika területén azért használják, mert finoman lehet interpolálni vele az időben [37]. Nem meglepő, hogy a nemlineáris identifikáció területén is megjelent [2].

Többféle spline-függvény is létezik, attól függően, hogy hányadfokú polinomok adják az alapját (leggyakrabban 2, 3 és 4), hogyan történik az interpoláció a szomszédos pontok között, illetve, hogy a kontrollpontokon átmegy-e a függvény, vagy sem.

Dolgozatomban a harmadfokú Catmull-Rom spline-függvényt használom, mely a többihez képest is számos előnnyel bír [38]. Egy példát mutat a 15. ábra.



15. ábra. Catmull-Rom spline-függvény.

A (Catmull-Rom) spline-függvények előnyei:⁵

- A memóriamentes nemlinearitások széles köre leírható a spline függvényekkel [4].
- A görbék simák, interpolálnak a kontrollpontok között [38].
- A függvény átmegy a kontrollpontokon, így közvetlen hatással van a görbe alakjára [38].
- Két kontrollpont közötti viselkedést négy kontrollpont határoz meg. Egy-egy kontrollpont csak a szűk környezetére van hatással [38].
- Egyszerűvé teszi a számítást az, hogy szakaszonként polinomokkal írható le [38].

A Catmull-Rom spline-függvények hátrányai:

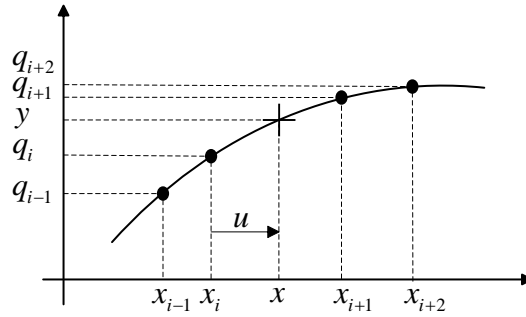
- Bár a kontrollpontoknál – ahol a polinomok összeérnek – az első deriváltak megegyeznek, a másodikak már nem [4].⁶

Egy spline-függvényhez elsőként meg kell határozni egy értelmezési tartományt, amit aztán kontrollpontokkal kell részintervallumokra osztani. Tegyük fel, hogy N darab x_i kontrollpontot helyezünk el az x -tengelyen úgy, hogy megfeleljen a (10) feltételnek.

$$x_0 < x_1 < \dots < x_{N-2} < x_{N-1} \quad (10)$$

⁵A felsoroltak egy része általánosan igaz az összes spline-függvényre.

⁶Például a B-spline-függvényeknél a második derivált is megegyezik, de ott a függvény nem megy át a kontrollpontokon.



16. ábra. Spline függvény és kontrollpontjai.

Figyelembe kell vennünk, hogy az első $[x_0, x_1]$ és az utolsó $[x_{N-2}, x_{N-1}]$ intervallumon nem értelmezett a függvény, mivel ezekre már nem teljesül, hogy négy kontrollpont között fekszenek. Amikor a spline-függvényt implementáljuk, a „kilógó” értékeket érdemes „visszanyesni”.

A szakirodalom általában egyenlő részekre osztja fel az értelmezési tartományt, mivel így könnyebb meghatározni azt, melyik részintervallumon kell az eredményt keresnünk. Én ezzel szemben nem egyenlő hosszúságú intervallumokkal dolgoztam, mivel úgy véltem, például tö-réspontok közelében érdemes sűrűbben elhelyezni a kontrollpontokat, és egyenes szakaszoknál ritkábban is lehet.

Tegyük fel, hogy az x pontban szeretnénk kiértékelni a spline-függvényt. Az eredményhez a következő lépésekkel jutunk el [2]. A megértésében segíthet a 16. ábra.

1. Meghatározzuk, melyik i intervallumon belül kell kiértékelni a függvényt (11).

$$i := \rightarrow x_i \leq x < x_{i+1} \quad (11)$$

2. Megállapítjuk, az intervallum hányadrészen helyezkedik el a pont (12).

$$u = \frac{x - x_i}{x_{i+1} - x_i} \quad 0 \leq u < 1 \quad (12)$$

3. Definiáljuk az \mathbf{u} és \mathbf{q} vektort illetve az \mathbf{M} mátrixot a (13), (14) és a (15) szerint. A \mathbf{q} vektorba az adott intervallumot határoló két és az azok mellett lévő egy-egy kontrollpont kerül.

$$\mathbf{u} = \begin{bmatrix} 1 & u & u^2 & u^3 \end{bmatrix} \quad (13)$$

$$\mathbf{q} = \begin{bmatrix} q_{i-1} \\ q_i \\ q_{i+1} \\ q_{i+2} \end{bmatrix} \quad (14)$$

$$\mathbf{M} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ -\frac{1}{2} & 0 & \frac{1}{2} & 0 \\ 1 & -\frac{5}{2} & 2 & -\frac{1}{2} \\ -\frac{1}{2} & -\frac{3}{2} & \frac{3}{2} & \frac{1}{2} \end{bmatrix} \quad (15)$$

A függvény értéke ekkor (16) szerint adódik.

$$f(x) = \mathbf{uMq} \quad (16)$$

Könnyen belátható, hogy az $f(x)$ \mathbf{q} -szerinti deriváltjának vektora \mathbf{uM} , amit gradiensek hívunk. Ez azért fontos eredmény, mert adaptív eljárásoknál ezt a \mathbf{q} vektor frissítésére (adott lépésben négy szomszédos kontrollpont frissítésére) használhatjuk. A későbbiek során ezzel a módszerrel fogok dolgozni.

4.2.3. A lineáris blokk modellezése

A lineáris szűrő megvalósításához kevesebb lehetőség közül választhatunk. Sjöberg et. al. [18] alapján mutatok be néhányat a lehetőségek közül, mivel igen jó áttekintést nyújt ezekről.

$$A(z)y(t) = \frac{B(z)}{F(z)}u(t) + \frac{C(z)}{D(z)}e(t) \quad (17)$$

Az egyenlet speciális esetei (a nem rögzített polinomok tetszőlegesen):

- Box-Jenkins modell, ha $A = 1$.
- ARMAX modell, ha $F = D = 1$
- Output-Error modell, ha $A = C = D = 1$.
- ARX modell, ha $F = C = D = 1$.

Dolgozatomban végig FIR szűrőt használok, mivel nincsen benne visszacsatolás. Ha a szűrő visszacsatolást tartalmaz, instabillá válhat, ami különösen kényes kérdés adaptív rendszerek esetén. Ennek ellenére gyakran IIR szűrővel valósítanak meg rendszereket, aminek az lehet az oka, hogy sokkal kevesebb paraméter beállítását igényli, mint egy FIR szűrő.

5. Kiegészítő eszközök nemlineáris rendszerek kezeléséhez

A ebben a részben olyan kiegészítő eszközöket mutatok be, melyekkel egy nemlineáris rendszer modellezéséhez és irányításához szükségesek.

Elsőnek arról írok, hogyan lehet a nemlinearitást mérni – illetve lehetséges-e ez egyáltalán. Ezután néhány olyan nemlineáris karakterisztika következik (kétféle telítődéses, dióda, négyzetes, köbös, holsáv és hídkapcsolás), melyek villamosmérnöki alkalmazásoknál előfordulhatnak. A következő részben lineáris, adaptív algoritmusokat ismertetek: az LMS-t, NLMS-t, annak egy változatát a JO-NLMS-t, és az RLS algoritmust. Végül két olyan struktúrát (XLMS, és ELMS) mutatok, mellyel egy nemlineáris rendszer linearizálni lehet.

5.1. „Mennyire nemlineáris?”

A nemlinearitás mértéke mindig relatív, mivel az alkalmazás dönti el, mennyire tartunk valamit nemlineárisnak. Ha egy olyan „rosszul tervezett” hídkapcsolásra gondolunk, melyben egyetlen változó ellenállás szerepel, hétköznapi alkalmazások esetén a nemlineáris karakterisztika ellenére nem tévedünk sokat. Precíz mérésre azonban már nem alkalmas.

Ugyanakkor jogosan merül fel az igény, hogy két hasonló szenzort vagy nemlinearitást össze tudjunk hasonlítani, és mértéket rendeljünk a megfigyelt jelenséghez. Ez alapján lehetne eldönteni, két nemlineáris rendszer közül melyiket alkalmazzuk, és kvantitatív eszközökkel ragadhatnánk meg egy rendszer öregedését is.

5.1.1. Egyenestől vett minimális, átlagos négyzetes eltérés

Emancipator és Kroll [39] úgy rendel a nemlineáris karakterisztikához egy mérőszámot, hogy kiszámítja, mekkora a legkisebb átlagos eltérés az ideális egyenestől. A szerzők szerint a négyzetes eltérést érdemes alkalmazni a (18) és a (19) képlet szerint, ahol $l(m, b)$ a négyzetes, átlagos eltérés, $f(x)$ a nemlineáris függvény, m az egyenes meredeksége, b az egyenes egyenletében a konstans, x_U a vizsgált intervallum alsó, x_L pedig felső határa.

$$L = \min_{m,b} \{l(m, b)\} \quad (18)$$

$$l(m, b) = \sqrt{\frac{\int_{x_L}^{x_U} (f(x) - mx - b)^2 dx}{x_U - x_L}} \quad (19)$$

Az (19) egyenletben a gyökvonás biztosítja, hogy a mérték azonos dimenzióban legyen a nemlineáris függvénnyel.

Ez a definíció semmit nem mond arról, hogyan kell megválasztani az alsó és felső határt. Mivel a valóságban a rendszerek bemenete általában korlátos, az értelmezési tartományának két végpontját tekinthetjük határnak. További problémát vet fel, általános esetben hogyan illesszük

a lineáris egyenest, mivel zárt képlet hiányában ez számításigényes folyamat lehet és erősen nemlineáris függvényeknél is félrevezető értéket adhat.

5.1.2. Itakura-Saito távolság

Gasparini et. al. [2] az Itakura-Saito távolságot használja arra, hogy a nemlinearitást egy számba sűrítse. Ennek a mértéknek folytonos és diszkrét változata is létezik, és azt ragadja meg, hogy mennyire különbözik a bemeneti jel spektruma a kimeneti jelétől. Gyakran beszédfelismerés esetén alkalmazzák.

$$D(\Phi, \Psi) = \left\langle \frac{\Phi}{\Psi}, 1 \right\rangle - \left\langle \log \frac{\Phi}{\Psi}, 1 \right\rangle - 1 \quad (20)$$

Az Itakura-Saito távolságot például a (20) szerint definiálhatjuk, ahol Φ és Ψ a bemenő és a kimenő jel spektruma. Ezzel megegyező más definíciók is léteznek, ezekről és a módszer korlátairól Enqvist [40] nyújt áttekintést.

5.1.3. Torzítási tényező

A torzítási tényezőt szinuszos jelek esetén alkalmazhatjuk. Ennek többféle definícióját is ismerik – én az IEEE által szabványosított formát (21) használom, mely azt mutatja, hogyan aránylik a jel felharmonikusainak teljesítménye a jel alapharmonikusának teljesítményéhez [41]. A (21) képletben az $X_{i,eff}$ az i . harmonikus effektív értéke.

$$THD = \frac{\sqrt{\sum_{i=2}^{\infty} X_{i,eff}^2}}{X_{1,eff}} \quad (21)$$

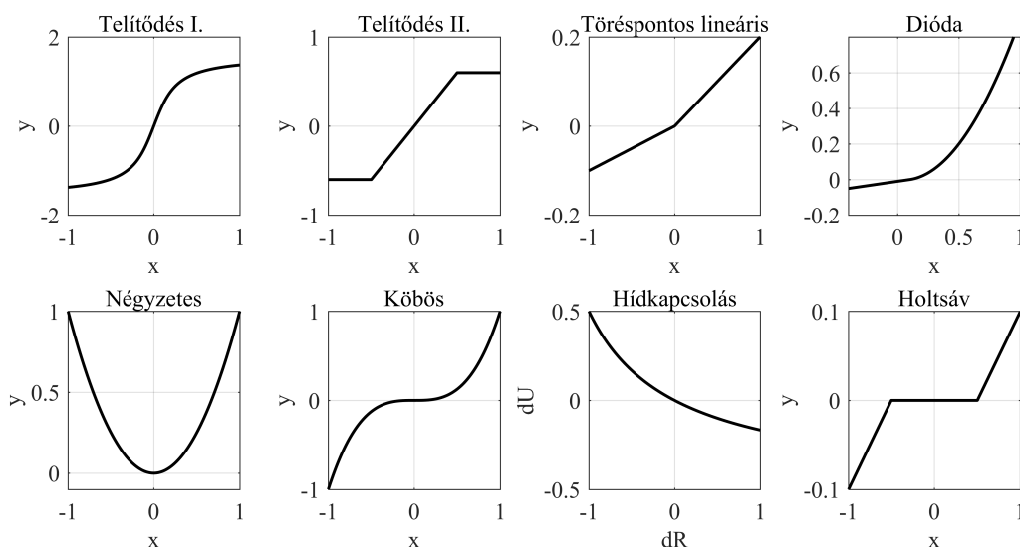
5.1.4. Koherenciafüggvény

A koherenciafüggvényt alkalmazhatjuk a be és kimeneti jelek közötti torzítás mérésére. A (22) képletben $S_x(f)$ és $S_y(f)$ az autoteljesítménysűrűség és $S_{x,y}(f)$ a kereszteljesítménysűrűség spektrum. Ezeket a Fourier-transzformáltak átlagolásával nyerhetjük.

$$C_{xy}(f) = \frac{S_x(f) S_y(f)}{S_{xy}^2(f)} \quad (22)$$

5.2. Gyakori nemlineáris karakterisztikák

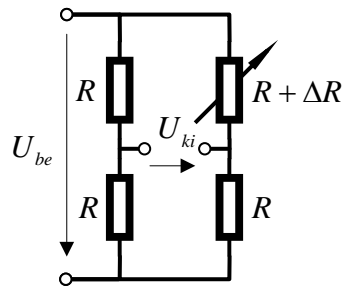
A későbbiek során az egyes modellezési technikák képességeit szimulációval vizsgálom, amihez elengedhetetlen volt, hogy olyan egyszerű és alapvető karakterisztikákat keressek, melyek a természetben vagy gyakorlati alkalmazásoknál előfordulnak. Terjedelmi korlátok miatt nyolcat választottam ki közülük: a telítődést, diódát, négyzetes és köbös függvényeket, a holsávot, hiszterézist,



17. ábra. Gyakori nemlineáris karakterisztikák.

és végül a hídkapcsolást. Ezeket mutatja a 17. ábra. Úgy gondolom ezek a függvények megfelelnek annak a célnak, hogy a modellek képességeit teszteljem. Érdekes még megjegyezni, hogy az itt bemutatott karakterisztikák túlzók lehetnek, de éppen ezért megfelelőek bizonyos jelenségek szemléltetésére:

- **Telítődés.** A telítődés azt jelenti, hogy hiába növeljük a bemenetet, a rendszer kimenete egy adott ponton túl nem lineárisan, hanem annál alacsonyabb ütemben növekszik tovább. Telítődéssel találkozunk DA-átalakítók kimeneténél, fizikai állapotváltozást igénylő folyamatoknál, és minden olyan helyzetben, amikor egy rendszer kimenete nem képes egy határ felé vagy alá menni. A telítődésnek kétféle változatát vizsgálom. Közöttük az a különbség, hogy a nemlineáris függvény törik-e, vagy sem. Előbbit szakaszonként lineáris függvénnyel, utóbbit *arctg* függvénnyel modelleztem.
- **Egy törésponttal rendelkező lineáris függvény.** Előfordulhat olyan helyzet, hogy a jel negatív értékeire a rendszer kimenete érzékenyebb, mint a pozitív értékekre. Ezt ragadja meg ez a karakterisztika.
- **Dióda-karakterisztika.** A dióda-karakterisztika azt a jelenséget írja le, amikor egy adott bemeneti határérték alatt a kimenet elhanyagolható mértékűre csökken, míg ha a bemenet meghaladja a határértéket, a kimenet lineárisan vagy exponenciálisan nő. A dióda-karakterisztikájának számos modellje ismert, melyek különböző feltételezésekkel élnek mind a zárótartománybeli, mind a nyitott állapotbeli viselkedését illetően. Az egyszerűség kedvéért olyan modellt használtam, mely zárótartományon egy kis pozitív meredekségű egyenessel írható le, nyitás után pedig a négyzetes függvénnyel. A görbe és az egyenes találkozásánál a deriváltak megegyeznek.



18. ábra. Hídkapcsolás.

- Négyzetes és köbös karakterisztika.
- „Rosszul tervezett” hídkapcsolás. Ha a hídkapcsolást nem jól tervezték, a bemenet és a kimenet között nemlineáris lesz a kapcsolat. Erre mutat példát a 18. ábra. A karakterisztikát erősen eltúloztam azért, hogy a karakterisztika jól látszódjon. Érdeemes megjegyezni, hogy egy kiegyenlített híd is lehet nemlineáris, mivel gyártani megegyező ellenállásokat szinte lehetetlen.
- Holsáv. A rendszer bizonyos bemenet esetén nem ad ki jelet, míg egy adott határ alatt és felett igen.

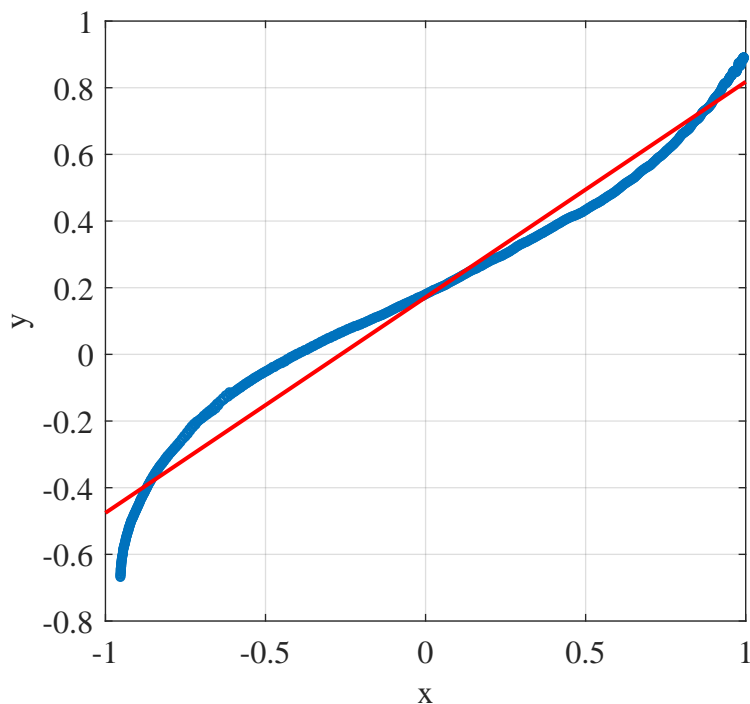
A nemlineáris függvényeken kívül érdemes megemlíteni a szignum függvényt és a hiszterézis jelenségét is. A szignum függvény a telítődés egy szélsőséges esete, de mivel egy ilyen rendszer nem kompenzálható, dolgozatomban nem vizsgálom. A hiszterézis tekercsek gyakori velejárója, kezelése azonban speciális eszközöket igényel, így kompenzálását dolgozatomban ugyancsak nem tárgyalom. Széleskörűen foglalkoznak azonban vele a szakirodalomban, például [42–44]. A Wiener-Hammerstein-modellel azonban a hiszterézis is megragadható [42, 44, 45].

Unbehauen [16] húsz gyakori nemlineáris függvényt említ, többek között a szakaszonként konstans és az abszolútérték függvényt (egyenirányító). A jövőben ezeket is érdemes lehet megvizsgálni.

5.2.1. Gyors becslő módszer gyengén nemlineáris, monoton növekvő karakterisztikák felmérésére

Vanbeylen [46] egy olyan lehetőségre is kitér, mellyel egy gyengén nemlineáris karakterisztikával rendelkező dinamikus rendszer nemlineáris függvényét közelítőleg meg lehet határozni.

Ez a technika nagyon hasznos lehet akkor, ha még semmilyen ismerettel nem rendelkezünk azon kívül, hogy a rendszer nemlineáris, és karakterisztikája monoton növekvő. (Ezért négyzetes karakterisztikát nem lehet vele felmérni.)



19. ábra. Shaker nemlineáris karakterisztikájának közelítő becslése. Pirossal egy lineáris regresszióval illesztett egyenest jelöltem.

Fontos, hogy az impulzusválasz amplitúdója legyen nagyságrendekkel kisebb a vizsgált jel-tartománynál, és valamilyen lassan (impulzusválasz lecsengéséhez képest) változó gerjesztésnél rögzítsük a kimenetet. A rögzített értékeket ezek után mind a bemeneti, mind a kimeneti jelnél növekvő sorba rendezzük, és kirajzoljuk az így kapott pontpárokat. (Lényegében ez két eloszlásfüggvény-szerű adatsort jelent.)

A dolgozat egy későbbi részében részletesebben vizsgálok és modellezek egy BRÜEL & KJÆR 4810 Mini-Shaker-t, melynek tulajdonsága, hogy alacsony frekvenciás gerjesztésnél nemlineáris viselkedést mutat.⁷

Ha például 30 Hz-es szinuszjelet adunk a bemenetére, és 48 kHz-es mintavételi frekvenciával rögzítjük egy rövid szakasz (fél periódus) kimenetét, a 19. ábrán látható karakterisztikát kapjuk. Ebből sejthető, hogy milyen nemlineáris viselkedésre számíthatunk.

5.3. Algoritmusok

Akár offline, akár online identifkálunk egy adaptív algoritmus alapvető építőköve egy-egy struktúrának [33]. Befolyásolja a konvergencia sebességét és a stabilitást is. A kiemelt szerep miatt

⁷Ennek okára is később térek majd ki.

bőséges szakirodalom vizsgálja, hogyan lehet jobbá tenni egy-egy technikát.

A következőket vehetjük figyelembe, amikor választunk közülük [47]:

- A szűrő struktúrája, mely nagyban befolyásolja a komplexitást.
- A konvergencia és követés sebessége. Az egyes módszerek eltérnek abban, mennyire gyorsan közelítik meg az optimális megoldást - amit a való életben általában nem érnek el. Gyakran teljesül, hogy gyorsabb konvergencia nagyobb végső maradó hibát eredményez.
- Számítással kapcsolatos megfontolások. Figyelembe kell venni, hogy a számábrázolás egyes rendszereken korlátos lehet, és törekedni kell arra, hogy a lehető legkevesebb erőforrást használjuk fel. Ezek erős kapcsolatban vannak a stabilitással, pontossággal és a zajtűréssel is.

Úgy gondolom, ezeken a megfontolásokon kívül érdemes olyan algoritmust keresni, amely zárt abban az értelemben, hogy nem igényli különféle paraméterek exogén - kézzel, előre törté-
nő - beállítását. Az a jó, ha az algoritmus maga, adaptívan állítja paramétereit. Az adaptív algoritmusokat részletesen körüljárja Isermann és Munchhof [26].

5.3.1. Az LMS algoritmus

Digitális jelfeldolgozás terén az LMS (*least mean squares*) az egyik legegyszerűbb, így legismer-
tebb és leggyakrabban alkalmazott algoritmus. Az LMS algoritmust kifejlesztője, Widrow [14]
alapján mutatom be.

Legyen \mathbf{X} a bementi jel aktuális és késleltetett értékeit tartalmazó oszlopvektor és \mathbf{W} a szűrő
együtthatóinak vektora. Az adaptív szűrő y kimenetét ekkor a két vektor szorzataként számítjuk
ki (23).

$$y_n = \mathbf{x}_n^T \mathbf{w}_n \quad (23)$$

Az e hibajelét megkapjuk, ha a kívánt d értékből kivonjuk a szűrő kimenetét (24).

$$e_n = d_n - y_n \quad (24)$$

A szűrő együtthatóit ezután a (25) szerint frissítjük, ahol μ a bátorsági tényező.

$$\mathbf{w}_{n+1} = \mathbf{w}_n + 2\mu e_n \mathbf{x}_n \quad (25)$$

A bátorsági tényező értéke tipikusan 0,001 vagy annál kisebb, nullához közeli szám – értéke az
együtthatók számától függ, gyakran az $\frac{1}{N}$ választással élnek. Minél nagyobb az értéke, annál
gyorsabban konvergál a szűrő, de annál nagyobb lesz a maradó hiba is. A bátorsági tényezőt
nem választhatjuk túl nagyra, mivel ekkor a szűrő instabillá válik. Ezért létezik egy optimális
értéke. A szűrő akkor konvergál, ha a bátorsági tényező értékére igaz a (26) összefüggés [14].

$$0 < \mu_n < \frac{1}{\mathbf{x}_n^T \mathbf{x}_n} \quad (26)$$

A konvergencia sebessége függ attól, hogyan viszonyul a bemenő jel autokorrelációs mátrixának legnagyobb és legkisebb sajátértéke egymáshoz [14, 47].

5.3.2. Az NLMS algoritmus és változatai

Az LMS algoritmus gyors konvergencia és a kis maradó hiba közötti ellentmondását úgy oldhatjuk fel, ha a bátorsági tényező értékét adaptívan állítjuk [8].

Az NLMS algoritmus ezt teszi: időben változtatja a bátorsági tényező értékét (27). A (27) egyenletben $0 < \tilde{\mu} < 2$ a sebességet befolyásolja, ϵ pedig egy 0-hoz közeli konstans, amivel egyrészt elkerülhető a 0-val való osztás, másrészt zaj esetén jobban teljesít a szűrő, bár zaj hiányában jelentősebb maradó hibához vezet [8].

$$\mu_n = \frac{\tilde{\mu}}{\mathbf{x}_n^T \mathbf{x}_n + \epsilon} \quad (27)$$

Az NLMS algoritmusnak számos továbbfejlesztése létezik, ezek adott környezethez igazodnak, és teljesítenek nagyon jól. Sawale és Yadav [48] olyan módszert fejlesztett ki, mely az NLMS-nél gyorsabban konvergál, ha megnövekedik a jel-zaj viszony (*SNR*). Módszerükben felhasználják a \mathbf{w} együttható vektor korábbi állapotait is, mivel ez csökkenti az együtthatók frissítésekor fellépő ingadozást, főként zajos környezet esetén (28).

$$\mathbf{w}_i = \frac{1}{N} \sum_{k=1}^N \mathbf{w}_{i-k} + \frac{\mathbf{x}_i^T}{\mathbf{x}_i \mathbf{x}_i^T} \left(d_i - \mathbf{x}_i \frac{1}{N} \sum_{k=1}^N \mathbf{w}_{i-k} \right) \quad (28)$$

$N = 1$ választással az egyszerű NLMS algoritmushoz jutunk. Bár a számítási kapacitás megnövekedik, a műveletek egyszerűek, és kevés erőforrást igényelnek.

Munkám során azt tapasztaltam, hogy az NLMS algoritmus nagyon kis jelek esetén nem teljesít jól, ezért kerestem egy olyat, melynél ez a probléma nem jelentkezik, és minden változót endogén módon állít be. Több lehetőséget is kipróbáltam, ezek közül Paleologu et. al. [8] megoldását – a JO-NLMS – algoritmust választottam. Kezdetben legyen az együtthatóvektor $\mathbf{w} = \mathbf{0}$, $m(0) = \epsilon > 0$ és $\hat{\sigma}_w^2(0) = 0$. Egy paraméterre van még szükség – a zaj teljesítményére – amit vagy ismerünk, vagy megbecsülhetünk futás közben is.

Az algoritmust a (29) egyenletek írják le.

$$\begin{aligned}
\hat{\sigma}_x^2(n) &= \frac{1}{N} \mathbf{x}^T(n) \mathbf{x}(n) \\
e(n) &= d(n) - \mathbf{x}^T(n) \mathbf{w}(n-1) \\
p(n) &= m(n-1) + N \hat{\sigma}_w^2(n-1) \\
q(n) &= \frac{p(n)}{N \hat{\sigma}_v^2 + (N+2)p(n) \hat{\sigma}_x^2(n)} \\
\mathbf{w}(n) &= \mathbf{w}(n-1) + q(n) \mathbf{x}(n) e(n) \\
m(n) &= (1 - q(n) \hat{\sigma}_x^2(n)) p(n) \\
\hat{\sigma}_w^2(n) &= \frac{1}{N} \|\mathbf{w}(n) - \mathbf{w}(n-1)\|^2
\end{aligned} \tag{29}$$

\mathbf{x} a bemeneti értékek a vektora, \mathbf{w} az együtthatóvektor, N az együtthatók száma. $\hat{\sigma}_x^2$ a bemenő jel, $\hat{\sigma}_w^2$ az együtthatók, $\hat{\sigma}_v^2$ a hibajel teljesítménye. e a hibajel, d a rendszer kívánt kimenete, m , p és q változók. Jobban megnézve az egyenleteket q bátorsági tényezőként is értelmezhető.

A szerzők hozzáteszik, hogy ha $\hat{\sigma}_w^2$ kicsi, lassabban konvergál a rendszer, de nagyon jó lesz a beállítás, ellenkező esetben ennek fordítottját tapasztalhatjuk. A hibajel teljesítménye a (30) szerint becsülhető.

$$\hat{\sigma}_v^2(n) = |\hat{\sigma}_d^2(n) - \hat{\sigma}_y^2(n)| \tag{30}$$

A kimeneti és a kívánt jel teljesítményét pedig érdemes fokozatosan állítani, a (31) egyenletek szerint, ahol λ tetszőleges pozitív 0 és 1 közötti konstans.

$$\begin{aligned}
\hat{\sigma}_d^2(n) &= \lambda \hat{\sigma}_d^2(n-1) + (1-\lambda) d^2(n) \\
\hat{\sigma}_y^2(n) &= \lambda \hat{\sigma}_y^2(n-1) + (1-\lambda) y^2(n)
\end{aligned} \tag{31}$$

5.3.3. Az RLS algoritmus

Az RLS algoritmus az LMS mellett a másik legelterjedtebb technika. Widrow [14] alapján mutatom be.

A módszer alapját a mátrix-inverziós lemma adja, ami megkönnyíti egy nagyméretű mátrix invertálását. (Az inverz közvetlen számítása rendkívül számításigényes lenne.) Az algoritmusban ezt az \mathbf{R} bemeneti autokorrelációs mátrix invertálásra használjuk fel. A cél, hogy ennek segítségével találjuk meg az együtthatóvektor optimális értékét, mely az inverzzel kifejezhető: $\mathbf{w}_{opt} = \mathbf{R}^{-1}p$, ahol p a bemenő és a kívánt kimeneti jel keresztkorrelációja.

Kezdetben az \mathbf{R}^{-1} mátrixot az egységmátrixhoz hasonlóan definiálhatjuk $\mathbf{R}_0^{-1} = \delta^{-1} \mathbf{I}$, ahol δ egy 1-hez közeli konstans.

$$\begin{aligned}
\mathbf{K}_n &= \mathbf{G}_n \mathbf{x}_n \\
\mathbf{G}_k &= \frac{\beta^{-1} \mathbf{R}_{n-1}^{-1}}{1 + \beta^{-1} \mathbf{x}_n^T \mathbf{R}_{n-1}^{-1} \mathbf{x}_n} \\
e_n &= d_k - \mathbf{w}_{n-1}^T \mathbf{x}_n \\
\mathbf{w}_n &= \mathbf{w}_{n-1} + \mu_n \mathbf{R}_{n-1}^{-1} e_n \mathbf{x}_n \\
\mu_n &= \frac{\beta^{-1}}{1 + \beta^{-1} \mathbf{x}_n^T \mathbf{R}_{n-1}^{-1} \mathbf{x}_n} \\
\mathbf{R}_n^{-1} &= \beta^{-1} \mathbf{R}_{n-1}^{-1} - \beta^{-1} \mathbf{K}_n \mathbf{x}_n^T \mathbf{R}_{n-1}^{-1}
\end{aligned} \tag{32}$$

Az algoritmust az (32) egyenletek foglalják össze. \mathbf{x} a bemenő jel értékeinek, \mathbf{w} az együtthatók vektora. e a hibát, d a rendszer kívánt kimeneti értékét jelöli. \mathbf{K} és \mathbf{G} egy sor sorvektor, illetve egy \mathbf{R} -rel megegyező méretű mátrix. β tetszőleges, 1-nél kisebb, de ahhoz közeli paraméter. Látszik, hogy a bátorsági tényező adaptívan változik, és a régi súlyvektorból számítjuk a hibát.

Az RLS nagyon gyorsan konvergál, ráadásul a konvergencia független a bemenő jel autokorrelációs mátrixa sajátértékeinek eloszlásától. Jól működik lassan változó jelek esetén is [47]. Az RLS hátránya azonban, hogy rendkívül számításigényes és nem általánosan stabil [14, 47]. Utóbbi tulajdonsága tapasztalatom szerint leginkább akkor jön elő, ha a β értéket csökkentjük (ekkor konvergál).

5.3.4. Az ELS algoritmus

A kiterjesztett legkisebb négyzetek módszere (*extended least squares, ELS*) egy olyan ARMAX modellen (33) alapul, mely a zajt és annak késleltetett értékeit is figyelembe veszi az adaptáció során [26]. Később erre a modellre még visszaalakok, azért mutatom most be.

$$y(k) = \psi^T(k) \theta(k-1) + e(k) \tag{33}$$

$$\begin{aligned}
\psi^T(k) &= \begin{pmatrix} -y(k-1), \dots, -y(k-m), \dots, \\ u(k-m), \dots, u(k-d-1), \dots, \\ \hat{v}(k-1), \dots, \hat{v}(k-p) \end{pmatrix} \\
\hat{\theta}(k+1) &= \hat{\theta}(k) + \gamma(k) \left(y(k+1) - \hat{\psi}^T(k+1) \hat{\theta}(k) \right) \\
\hat{v}(k) &= y(k) - \psi^T(k) \hat{\theta}(k) \\
\gamma(k) &= P(k+1) \psi(k+1) \\
P(k) &= \left(\psi^T(k) \psi(k) \right)^{-1}
\end{aligned} \tag{34}$$

Az algoritmust a (34) mutatja be. ψ a y kívánt kimeneti, u bemeneti és v becsült zaj értékekből összeállított vektor. $\hat{\theta}$ az együtthatók vektora, \hat{v} a zaj becslője, γ és P egy számított paraméter, illetve mátrix. γ a korrekciós vektor.

Az ELS algoritmus hátránya, hogy torzított becslést ad, ha a zaj modellje nem pontos, a zajt leíró paraméterek lassabban konvergálnak, mint a többi paraméter és előre meg kell határozni a zajt leíró paraméterek számát. Előnye azonban, hogy a zajt is becsli [26].

5.4. A paraméterek illesztése

A paraméterek illesztésének az a célja, hogy megkeressük a rendszerre legjobban illeszkedő modellt. Ezt nehezíti, hogy a nemlineáris rendszer belső változói általában nem megfigyelhetők [45]. A nemlineáris rendszerek nagy száma és karakterisztikáik sokszínűsége miatt nincsen olyan megközelítés, amivel a nagy többséget identifikálni lehetne. Ebből fakadóan rengeteg különféle technika született, melyekről Isermann és Munchhof [26], illetve Nelles [17] kitűnő áttekintést nyújt.

A kiválasztott modell-struktúra általában leszűkíti a paraméter-becslő eljárások körét:

- A nemlineáris modellek egy jelentős része ugyanis paramétereiben lineáris, vagy sokszor a nemlinearitás invertálásával azzá tehető [17]. Ilyen a Volterra és a Hammerstein-modell⁸ is [5]. Ez a tulajdonságuk nagyban leegyszerűsíti az illesztést, mert az összes – elsősorban lineáris rendszerekre kifejlesztett – algoritmust felhasználhatjuk, feltéve, hogy négyzetes költségfüggvényt alkalmazunk [17]. Választhatunk direkt (LS), rekurzív (RLS) és iteratív (LMS) technikák közül.

Mivel a nemlineáris rendszerek gyakori jellemzője, hogy a sajátértékeinek szórása nagy, azoknál a módszereknél, melyek a sajátértékek eloszlásától függenek (például az LMS) lassú lesz a konvergencia [5].

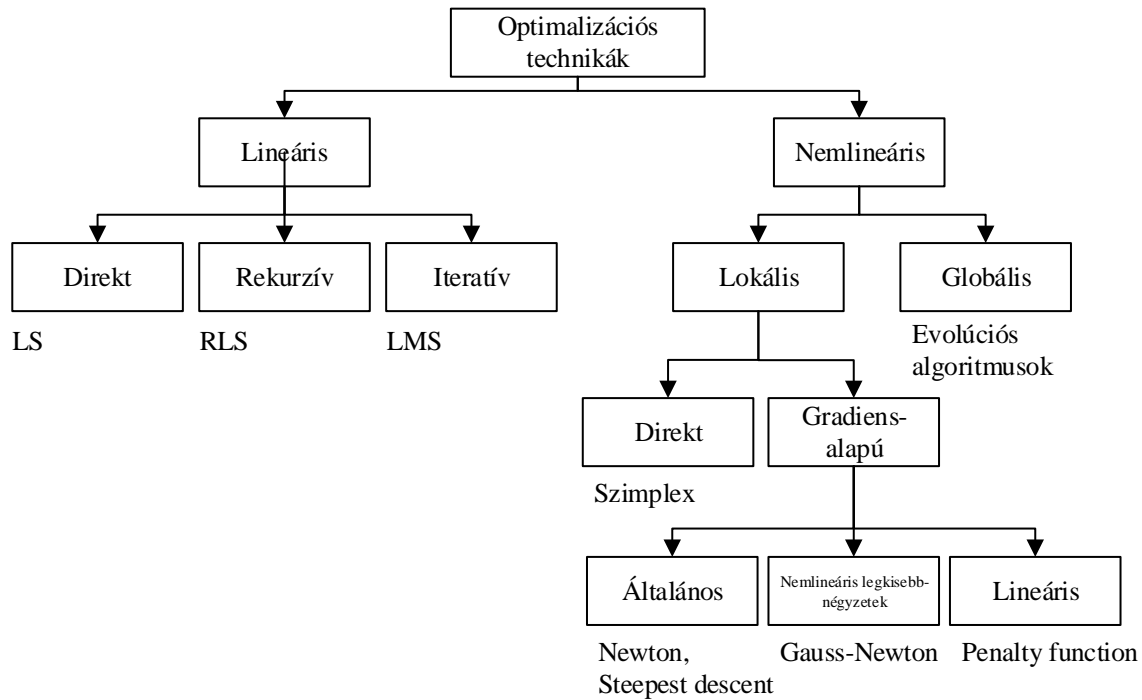
- A paramétereikben nemlineáris modellek – mint például a Wiener-modell – komplexebb megközelítést igényelnek (ELS) [26]. A következő problémák léphetnek fel:
 - Sok lokális minimum létezik [17]. Nagy valószínűséggel ilyen találunk, ha a paramétereknek nem megfelelő kezdeti értéket adunk [9,18].
 - Nincsen analitikus megoldás, csak iteratív megközelítéssel lehet élni [17].
 - Nagyon ritkán lehet online identifikálni őket [17].

A módszerek egy része lokális optimumot keres, a másik része globálisat (evolúciós algoritmusok). A lokális minimum keresése történhet direkt (nem szükséges gradienst számítani) és gradiens alapú algoritmusokkal.

Ezt a csoportosítási lehetőséget foglalja össze a 20. ábra.

Egy másik kategorizálást mutat be Isermann [26]:

⁸Ehhez speciálisan kell megválasztani a paramétervektort. Ha módszerünk nem biztosítja, hogy a paramétervektor és az adatvektor skaláris szorzataként megkapjuk a teljes Hammerstein-modell kimenetét (ami például polinomokra és spline-okra teljesül), más paraméterillesztő technikát kell keresnünk.



20. ábra. Modellillesztő technikák csoportosítása [17] alapján.

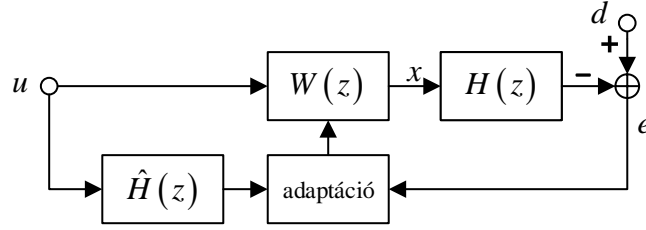
- Nem rekurzív paraméter becslő módszerek: ide tartozik az LS, a GLS, az ELS és a maximum-likelihood módszer is.
- Rekurzív paraméterbecslő módszerek: ide tartozik az NLMS és az RLS algoritmus.
- Egyéb módszerek: ide tartoznak például a neurális hálók.

Ezenkívül az eljárásokat csoportosíthatjuk aszerint, hogy idő- vagy frekvencia-tartományban működnek, iteratívak vagy sem. Egy ilyen másik csoportosítást mutat [17].

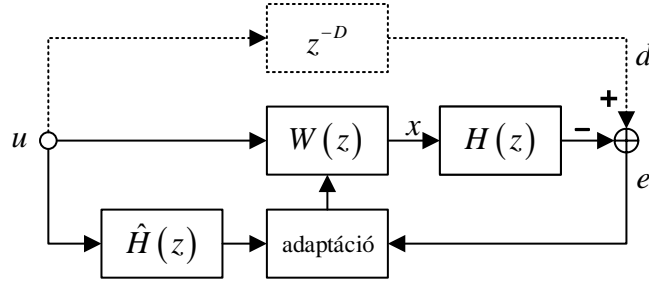
A paraméterek adaptív meghatározásakor az adaptációs hurok okozhat instabilitást, de erre is született megoldás a szakirodalomban.

5.4.1. Az XLMS algoritmus és kompenzálásra használható módosítása

Az XLMS algoritmus inverz online meghatározására használható technika, ami garantálja az adaptációs hurok stabilitását (21. ábra). A szakaszt $H(z)$, a szakasz modelljét $\hat{H}(z)$, az adaptív szűrőt $W(z)$ jelöli. A kívánt kimeneti jel d , a bemenő jel u , az adaptív szűrő kimenete x , a hibajel e , végül a D lépéssel történő késleltetés z^{-D} . Az algoritmus felhasználja a rendszer offline identifikált modelljét. A modellnek annyira kell pontosnak lennie, hogy a fázishiba ne haladja meg a $\frac{\pi}{2}$ -t, mivel ellenkező esetben a rendszer instabillá válik.



21. ábra. Az XLMS algoritmus. [15] alapján.



22. ábra. Az XLMS algoritmus kompenzálásra használható módosítása, melyet szaggatott vonal jelöl.

A hibajelét úgy kapjuk, hogy a szakasz kimeneti jelét összehasonlítjuk a kívánt kimeneti értékkel (35).

$$e_n = d_n - H(z)x_n \quad (35)$$

$$e_n = z^{-D}u_n - H(z)x_n \quad (36)$$

A súlyok a (37) szerint frissülnek.

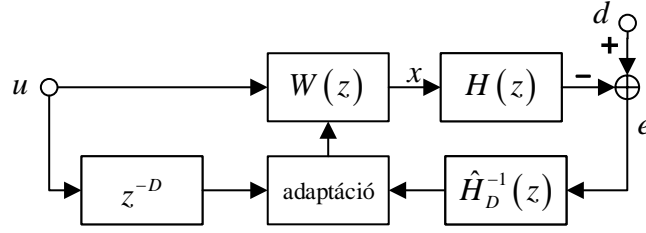
$$w_{n+1} = w_n + \mu e_n \hat{H}(z)u_n \quad (37)$$

A struktúra biztosítja azt, hogy az adaptáció során a megfelelő hibajel-érték mellé a megfelelő bemenetijel-érték kerüljön. Bár az XLMS algoritmus stabil, néhány esetben rendkívül lassú lehet a beállása [15].

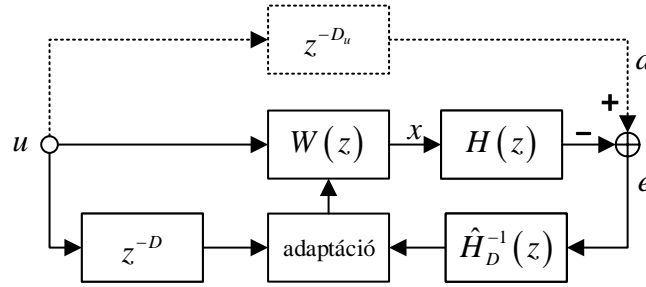
Speciális esetben, ha az XLMS algoritmust kompenzálásra használjuk, a kívánt kimeneti hely szerepét a késleltetett bemenet veszi át, amit a 22. ábrán szaggatott vonallal jelöltem. A D késleltetésnek meg kell haladnia a kontroller és a szakasz együttes késleltetését.

5.4.2. Az ELMS algoritmus és kompenzálásra használható módosítása

A megfelelő bemeneti és a hozzájuk tartozó hibajelértékek párosítását másként is megoldhatjuk, például úgy, hogy a hibajel a szakasz $\hat{H}_D^{-1}(z)$ késleltetett inverzén halad át és így találkozik



23. ábra. Az ELMS algoritmus. [15] alapján.



24. ábra. Az ELMS algoritmus kompenzálásra használható módosítása.

a bemeneti jel $z^{-D}u$ késleltetett értékeivel [15]. Ennek a struktúrának ELMS a neve, és a 23. ábrán látható. A késleltetett inverz modelljét offline, analitikusan kell megalkotnunk a rendszer offline kialakított modelljéből.

A hibajelel úgy kapjuk, hogy a d kívánt értéket összevetjük a szakasz kimenetével (38).

$$e_n = d_n - H(z)x_n \quad (38)$$

$$e_n = z^{-D_u}u_n - H(z)x_n \quad (39)$$

LMS algoritmusnál a súlyokat a (40) szerint adaptáljuk.

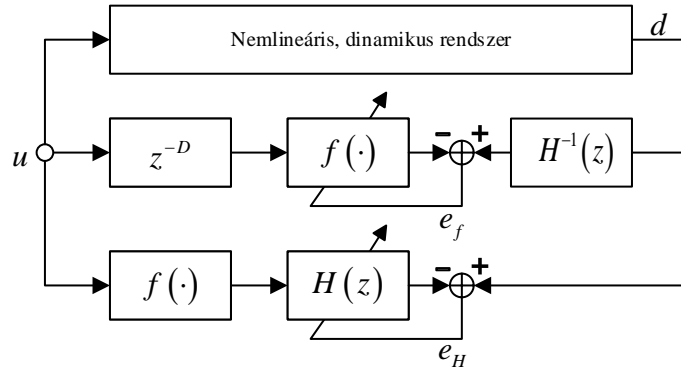
$$w_{n+1} = w_n + \mu \hat{H}_D^{-1}(z) e_n z^{-D} u_n \quad (40)$$

Az ELMS algoritmus addig képes követni a rendszer lassú változását, amíg a késleltetett inverz modellje megfelelően gondoskodik a helyes fázisról az adaptációs hurokban.

Speciális esetben, ha az ELMS algoritmust kompenzálásra használjuk, a kívánt kimeneti érték megegyezik a késleltetett gerjesztéssel (39). Ezt a 24. ábrán szaggatott vonallal jelöltem.

5.4.3. Paraméterek illesztése Hammerstein-modellhez

A Hammerstein-modell, ha FIR szűrőt és spline-t tartalmaz paramétereiben lineáris modell. Ebből az következik, hogy paramétereit LMS, RLS, vagy ezekhez hasonló adaptív gradiens alapú



25. ábra. Struktúra Hammerstein-modell illesztéséhez.

megközelítéssel taníthatjuk. (A spline függvény gradiense – ahogyan korábban írtam – megegyezik az \mathbf{uM} vektorral.)

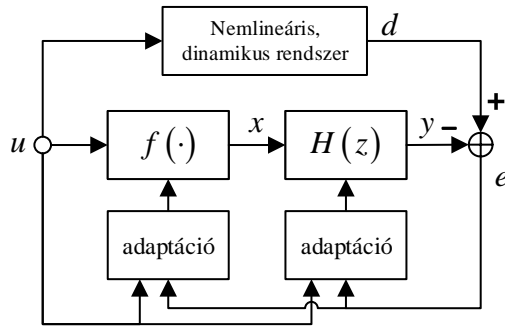
A Hammerstein-modellnél a paraméterek illesztését azonban jelentősen megnehezíti az, hogy:

- A struktúra erősen leegyszerűsíti a valóságot [33].
- A belső, két blokk közötti változó (a statikus nemlinearitás kimenete) közvetlenül nem megfigyelhető [33, 49].
- A két blokk bármelyikében lehet erősítés [33]. Például ha a nemlineáris karakterisztikát a -val szorozzuk, a lineáris dinamikus szűrőben oszthatunk is vele, és így elvileg számos jó megoldást kapunk.
- A hibajel visszacsatolását úgy kell megoldani, hogy a megfelelő bemeneti jelhez a megfelelő hibajel kerüljön.

Az erősítés problémáját úgy oldhatjuk meg, hogy egy paramétert rögzítünk [2]. Ha a lineáris szűrőt IIR szűrőként valósítják meg, gyakran a $b_0 = 1$ feltevéssel élnek. Másik megoldás, ha kimérjük a nemlineáris rendszer d késleltetését és a FIR szűrő esetén a $w_d = 1$ választással élünk [2]. Utóbbihoz elengedhetetlen, hogy a rendszer késleltetését mérni tudjuk, ami a gyakorlatban több problémát is felvet.

A hibajel visszacsatolása jelentős probléma, és úgy láttam, hogy a szakirodalom nem foglalkozik elég mélyrehatóan azzal, mi történik, ha a szakasznak jelentős a késleltetése. Azt tapasztaltam, hogy ha a hibajelet közvetlenül csatoljuk vissza, a nemlineáris függvény képtelen rendesen adaptálódni. Nem jó megoldás az előbb említett technika sem, miszerint a rendszer késleltetését mérjük.

Ezért olyan struktúrát kellett találni, amivel a feladat megvalósítható. Induljunk ki abból, hogy adott a nemlineáris késleltetést tartalmazó rendszer és a Hammerstein-modell. Közös u jellel gerjesztjük őket és megfigyeljük a nemlineáris rendszer d kimenetét.



26. ábra. Hammerstein-modell helytelen visszacsatolással.

A lineáris szűrő tanításához a Hammerstein-modell és a rendszer kimenete közötti hibajelet közvetlenül visszacsatolhatjuk. Ezzel a lineáris szűrő megbecsülheti a rendszer késleltetését. A nemlineáris függvény tanításához úgy párosítjuk eközben a megfelelő bemeneti és hibajel-értékeket, hogy a modell lineáris blokkjának kiszámítjuk az inverzét, amin átvezetjük a rendszer kimenetét, és ezt hasonlítjuk össze azzal a jellel, amit a nemlineáris blokk kiad abban az esetben, ha az inverz hosszának $(2D)$ felével (D) késleltethetjük a bemeneti jelet. A következő fejezetben vizsgálom meg részletesen ennek a struktúrának a tulajdonságait.

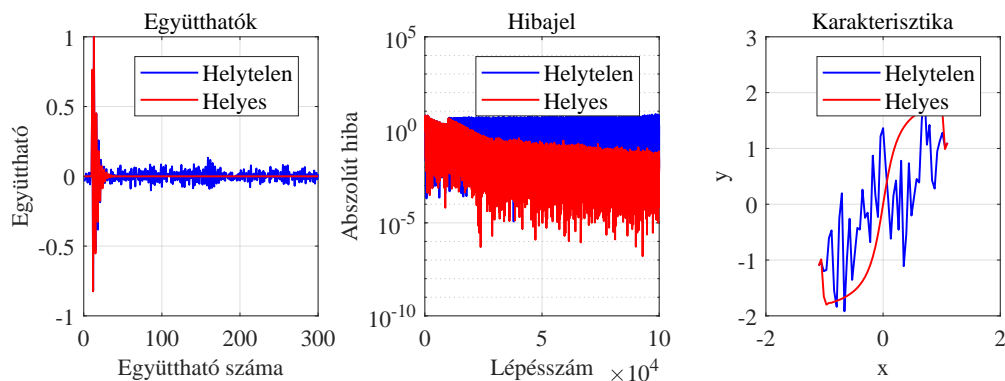
Tegyük fel, hogy a Hammerstein-modellnek megfelelően kötünk össze egy statikus nemlineáris és egy lineáris dinamikus blokkot. Előbbihez Catmull-Rom spline-függvényt, utóbbihoz FIR szűrőt használunk. Modellezni szeretnénk egy olyan nemlineáris rendszert, mely a szimuláció során Wiener-modell szerint tartalmazza a (41) rendszert⁹ és az $\arctang(3x)$ nemlineáris karakterisztikát. Identifikációhoz a 26. ábrán látható struktúrát használjuk. LMS algoritmussal, egy időben adaptáljuk az együtthatókat. A gerjesztés véletlen, normális eloszlású, fehér zaj. Az első 10 ezer lépésben csak a lineáris szűrőt tanítjuk, utána azonban egyszerre mindkettőt. A szűrő együtthatóinak száma 300 a spline-kontrollpontok száma 51.

$$H(z) = z^{-10} \frac{z^2 - 0,4164z + 1,2346}{z^2 + 0,6627z + 0,6414} \quad (41)$$

A hibajel alakulását mutatja a 27. ábra kék színű grafikonja. Az első 10 ezer lépésben a hibajel csökken – ekkor csak a lineáris blokk adaptálódik. Amikor azonban bekapcsol a nemlineáris blokk, a hibajel hitelen megugrik, és közel állandó szinten marad. Ennek a jelenségnek az oka az, hogy a rendszer késése miatt a bemeneti jel értékei nem a megfelelő hibajelértékekkel találkoznak az LMS algoritmussal. Ez a lineáris szűrő tanításánál nem okoz gondot, de a statikus nemlinearitásnál már igen.

Megoldást jelenthet, ha megmérjük a nemlineáris rendszer késleltetését, és ennek megfelelően késleltetjük a bemeneti u jelet, mielőtt a nemlineáris blokkot adaptálnánk. Ennek a módszer meg lehetőségesen körülményes.

⁹Ez a rendszer Sujbert [15] példa-szakasza, késleltetéssel kiegészítve.



27. ábra. Hammerstein-modell hibajelének alakulása helyes(piros) és helytelen (kék) visszacsatolás esetén

Helyette érdemesebb egy olyan struktúrát alkalmazni, mint ami a 25. ábrán látszik. A struktúra a következőképpen működik:

1. A Hammerstein-modell kimenetét úgy kapjuk, hogy a bemenő u jelet átvezetjük az $f(\cdot)$ nemlineáris blokkon majd a $H(z)$ lineáris dinamikus szűrőn.
2. A $H(z)$ szűrőt közvetlenül tanítjuk azzal a visszacsatolt e_H hibajellel, melyet a rendszer d és a Hammerstein-modell kimenete közötti különbségként kapunk.
3. Analitikusan meghatározzuk a $H(z)$ inverzét. Az inverz szűrő hossza $2D$. Ezen átvezetjük a rendszer kimeneti jelét.
4. Az u bemeneti jelet késleltetjük D lépéssel, majd átvezetjük a nemlineáris blokk másolatán. A blokk kimenet és a $\hat{H}(z)$ inverz kimenete közötti e_f hibajellel adaptáljuk a nemlineáris blokkot. Az adaptáció eredményét azonnal átmásoljuk.

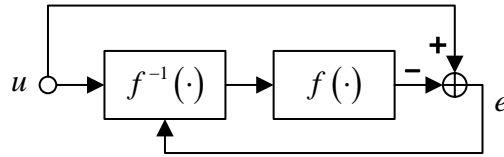
Ha az előbb bemutatott kísérletet ezzel a struktúrával végezzük el, a 27. ábrán pirossal rajzolt hibajelét, impulzusválaszt és nemlineáris karakterisztikát kapjuk.

5.4.4. Inverz meghatározása frekvencia-mintavételi eljárással

Tegyük fel, hogy egy rendszert a Hammerstein-modell szerint identifkálunk. Ha a nemlinearitást és a lineáris szűrőt is külön-külön invertáljuk, linearizáljuk a rendszert.

A lineáris szakasz invertálására egy lehetséges módszer a frekvencia-mintavételi eljárás. Legyen az eredeti impulzusválasz $h(n)$. Ekkor a lépések a következők:

1. A $h(n)$ diszkrét Fourier-transzformálásának eredménye $H(z)$. Ezt megfelelően sok pontban kell kiszámítani, sokkal több pontban, mint $h(n)$ hossza.
2. Pontonként kiszámítjuk a $H(z)$ inverzét: $G(z) = \frac{1}{H(z)}$.



28. ábra. Invertálható függvény adaptív invertálása.

3. Meghatározzuk $G(z)$ inverz diszkrét Fourier-transzformáltját. Az eredmény $g_0(n)$ tartalmazhat olyan komplex számokat, melyeknek kis képzetes részük is van, ezért érdemes az inverz transzformáció abszolút értékét is venni. Normális körülmények között ha $h(n)$ valós volt, $g_0(n)$ csak valós résszel rendelkezik.
4. $g_0(n)$ első fele „lecseng”, vége „felcseng”. $g(n)$ úgy képezendő, hogy megfelelően hosszú részt vágunk ki a végéről (ez határozza majd meg a késleltetést), amit az elejéből kivágott megfelelően hosszú rész elé illesztünk.

Az eljárás során figyelembe kell venni, hogy az inverz impulzusválasza sokkal hosszabb is lehet, mint az eredeti impulzusválasz. Egy nemlineáris függvényt például a 28. ábrán látható eljárással invertálhatunk adaptív módon – vagy megtehetjük azt analitikusan is. A függvény inverzére teljesülnie kell, hogy $f(f^{-1}(x)) = x$.

5.4.5. Inverz közelítő meghatározása analitikusan

Widrow [14] alapján, némi kiegészítéssel és részletesebb magyarázattal bemutatom, hogyan lehet meghatározni a nem minimálfázisú szakaszok késleltetett inverzét, ha ismerjük annak átviteli függvényét. Ehhez felhasználok Sujbert [15] példa-szakaszát, mely a (42) alakot ölti.

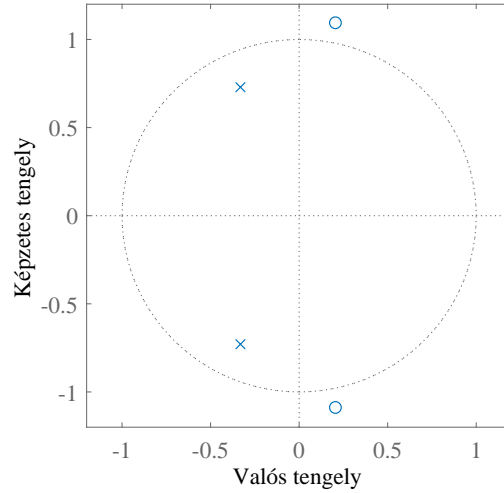
$$H(z) = \frac{z^2 - 0,4164z + 1,2346}{z^2 + 0,6627z + 0,6414} \quad (42)$$

A diszkrét idejű átviteli függvényhez a 29. ábra látható pólus-zérus térkép tartozik. A szakasz zérusai az egységkörön kívül helyezkednek el. Az átviteli függvény inverzéhez úgy jutunk, hogy felcseréljük a pólusokat és a zérusokat egymással (43).

$$H^{-1}(z) = \frac{z^2 - 0,6627z + 0,6414}{z^2 - 0,4164z + 1,2346} \quad (43)$$

A (43) kifejezést kétféleképpen is sorba fejthetjük, egyrészt a nem kauzális, de stabil forma, a Taylor-sor (44), másrészt a kauzális, de nem stabil Laurent-sor (45) szerint. A Taylor-sorba fejtt alakot kauzális szűrővel nem tudjuk modellezni.

$$\hat{H}^{-1}(z) = 0,5195 + 0,3616z + 0,2672z^2 - 0,3829z^3 + \dots \quad (44)$$



29. ábra. A $H(z)$ rendszer pólus-zérus térképe.

$$\hat{H}^{-1}(z) = 1 + 0,2463z^{-1} - 0,6958z^{-2} - 0,0144z^{-3} + \dots \quad (45)$$

Tegyük egy z^{-d} késleltetést a rendszerbe úgy, hogy a $d = 3$ választással élünk (46) és (47). Ha nagyobbra választjuk a d értéket, akkor több tag is megmarad, és pontosabb lesz az eredmény.

$$z^{-d}\hat{H}(z) = z^{-d}(0,5195 + 0,3616z + 0,2672z^2 - 0,3829z^3 + \dots) \quad (46)$$

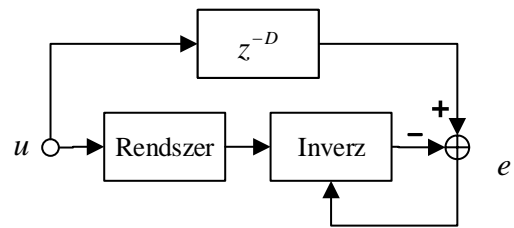
$$z^{-3}\hat{H}(z) \simeq 0,5195z^{-3} + 0,3616z^{-2} + 0,2672z^{-1} - 0,3829 \quad (47)$$

Ezzel egy stabil, közelítő inverzéhez jutunk az eredetileg nem minimálfázisú szakaszunk.

5.4.6. Inverz meghatározása adaptív módszerrel

Korábban megjegyeztem, hogy kétféle megközelítéssel lehet kompenzálni egy nemlineáris, dinamikus rendszert: az inverzét vagy sorosan elé, vagy sorosan mögé kötjük. A dolgozatban azokat a technikákat vizsgálom, melyekkel fizikai (akusztikus, mechanikai) kimenetű rendszereket lehet kompenzálni. Erre csak akkor vagyunk képesek, ha a szabályzót a rendszer elé kötjük. Azt is kiemeltem, hogy ebben az esetben a hibajel közvetlen visszacsatolása – néhány speciális esettől eltekintve – instabillá teheti a rendszert, ezért van szükség például az XLMS vagy ELMS algoritmusra.

Ha azonban a szabályzó a rendszer mögé kerül, a hibajel visszacsatolása nem okoz instabilitást, és az így kialakított adaptív struktúra minden esetben stabil lesz [15]. Ezt az elrendezést a 30. ábra szemlélteti.



30. ábra. Késleltetett inverz adaptív, minden esetben stabil meghatározása.

6. Gyakorlati tapasztalatok – az egyszerűsített Volterra-modell

Ebben a részben megvizsgálom, hogyan hatnak különféle paraméterek a Volterra-modell teljesítményére. Azokról a következtetésekről, melyeket ebben a fejezetben levonok, az általam megismert szakirodalom nem tesz említést.

A modellnek azt az egyszerűsített változatát használom, amelynél a bemeneti értékek hatványai szerepelnek.

A szimulációk során a rendszert egy olyan Hammerstein-modell jeleníti meg, melynek a lineáris dinamikus blokkja a korábban már bemutatott (41) átviteli függvényvel jellemezhető. Nemlineáris karakterisztikák közül többet is kipróbálok majd.

Fontosnak tartottam, hogy megszerzett ismereteimet egy kézzel fogható valóságos eszközön is kipróbáljam. Választásom a BRÜEL & KJÆR 4810 Mini-Shaker-re esett, mivel sok vibroakusztikai mérés alapul rajta, de alacsony frekvenciás gerjesztésnél (30 Hz környékén) nemlineáris viselkedést mutat (19. ábra). Ennek oka az, hogy a shakernek azonos gyorsuláshoz, alacsony frekvencián jobban ki kell térnie az egyensúlyi helyzetétől, mint magasabb frekvenciákon. A kitérés visszatartó ereje – egy membrán – azonban nagyobb kitérésnél exponenciálisan nagyobb (nemlineáris) erőt fejt ki a mozgórészre – ehhez adódnak hozzá további nem ideális hatások (például az asztal rezgései). A jelenség szinuszos gerjesztésnél jól megfigyelhető. Ezt majd a fejezet egy későbbi részében bemutatom. A szakirodalomban nem találtam olyan munkát, mely egy shaker kompenzálásával foglalkozott volna, így munkámat nem tudtam más eredményeihez hasonlítani.

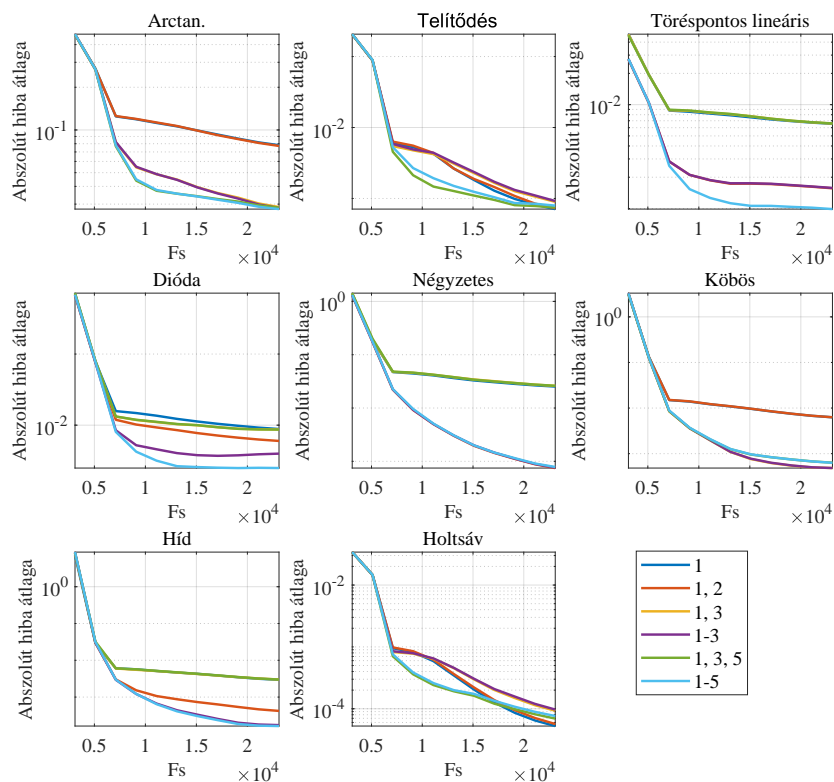
6.1. A bemenő jel szűrése

Korábban megjegyeztem, hogy a Volterra-modellnél a magasabb rendű hatványok jelenléte nem megfelelően magas mintavételi frekvencia esetén a komponensek átlapolódásához és így a szűrő rendellenes működéséhez vezet. Ha n a legmagasabb hatvány, és a bemenő gerjesztés felső sávhatára f_{max} , akkor ahhoz, hogy biztosan ne történjen átlapolódás, a mintavételi frekvenciának meg kell haladnia a $2nf_{max}$ küszöbértéket.

Volterra-modell esetén így ügyelni kell a hatványok számának és a mintavételi frekvenciának a helyes megválasztására. Ha a bemenő gerjesztés fehér zaj, azt a hatványok számától függően aluláteresztő szűrővel szűrni kell. Az általam feldolgozott szakirodalomban nem találtam erre vonatkozó utalást, de a jelenséget kísérlettel igazolni tudtam.

A 31. ábrán azt láthatjuk, hogyan alakul a 7000. és a 10000. lépés közötti hibajel abszolút értékének átlaga nyolc nemlineáris karakterisztikánál, ha növeljük a mintavételi frekvencia értékét. A hatást többféle bemenő hatvány-kombinációnál is megvizsgáltam: egyenként szimuláltam az $\{1\}$, $\{1-2\}$, $\{1, 3\}$, $\{1-3\}$, $\{1, 3, 5\}$ és a $\{1-5\}$ lehetőségeket.

A szimuláció elrendezését a 32. ábra mutatja. A szűrő együtthatóinak száma hatványonként 80 volt, és a gyors konvergencia érdekében RLS algoritmust használtam. Az egyenletes

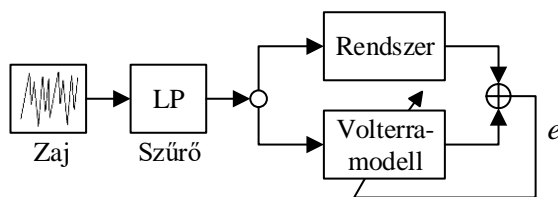


31. ábra. Volterra-modell hibajelének mintavételi frekvenciától függő alakulása RLS algoritmus esetén

–1 és 1 közötti véletlen eloszlású zajnál az algoritmus stabil volt. A gerjesztést a MATLAB DSP Toolbox-ának aluláteresztő szűrőjével szűrtem: a szűrő fokszáma 100, sávhatára 1500 Hz, zárótartománybeli elnyomása 90 dB.

A kísérletben az 5.2. alpontban bemutatott, 17. ábrán látható nemlineáris karakterisztikákat használtam fel. A 31. ábrára tekintve rögtön szembetűnik, hogy a 3 kHz-nél még igen jelentős hiba nagymértékben csökken, míg a mintavételi frekvencia el nem ér 7 kHz környékére. Innentől a karakterisztikától és a hatványok számától függően csökken tovább:

- Az arctan függvény Taylor-sorában csak páratlan hatványok szerepelnek. Így nem meglepő,



32. ábra. Szimuláció elrendezésének blokkvázlata

hogy a párosak nem járulnak hozzá a hibajel csökkenéséhez. Az 1. és 2. hatvány együtt ugyanúgy teljesít, mintha csak az első hatványt használnánk fel, és ez igaz az 1 – 3. és az 1 – 5. hatványokra is: nem romlik a modell pontossága, ha kivesszük a párosakat. A hibajeleket tekintve az 1 és az 1 – 2. hatványok szerepelnek a legrosszabbul. Érdekes, hogy amint a mintavételi frekvencia a 25 kHz-hez közelít, a többi hatvány azonosan teljesít, azonban 7.5 – 25 kHz között jobb modellt kapunk, ha több hatványt használunk fel.

- Ha a telítődéses karakterisztika élesen megtörik, 25 kHz-nél közel azonos hibát kapunk mind a hat esetben. 7.5 és 25 kHz között az $\{1, 3, 5\}$ hatványt tartalmazó modell a legpontosabb. Ezt követi az 1 – 5. hatványt tartalmazó modell – ebből azt a következtetést lehet levonni, hogy a páros tagok itt rontanak a modellen.
- Ha a karakterisztika két olyan pozitív meredekségű egyenesből áll, melyek az $x = 0$ pontban törnek, a hat lehetőség közül akkor kapjuk a legjobb modellt, ha az első öt hatvány mindegyikét felhasználjuk. Az 1 – 2. és az 1 – 3. hatványos modell hibajele azonos, és jobban teljesít, mint az 1., az $\{1, 3\}$ és az $\{1, 3, 5\}$ hatványt tartalmazó modell. Ezek szerint egy ilyen karakterisztika esetén a páratlan hatványok nem javítanak a modellen.
- Egy dióda modellezésénél érdemes a lehető legtöbb hatványt felhasználni, de mivel az 1 – 3. hatványok hatékonyabbak, mint a többi (a 1 – 5. hatványt kivéve) és az 1, 3, 5. hibája megegyezik az 1, 3. hatvány hibájával, ha a számítási kapacitás véges, érdemes inkább egy alacsonyabb páros hatványt betenni egy magasabb páratlan helyett.
- Előzetesen azt várhatjuk, hogy a négyzetes karakterisztika hibajele akkor lesz a legkisebb, ha a 2. hatványt tartalmazza a Volterra-sor. Ennek megfelelően alakult a kísérlet.
- Ha köbös függvényt modellezünk, az 1. és az 1 – 2. hatvány azonos eredményre vezet. Kisebb lesz a hiba, ha az 1 – 5. vagy az $\{1, 3, 5\}$ hatványokat választjuk. Legjobban akkor járunk, ha csak az $\{1, 3\}$. vagy az 1 – 3.-at használjuk fel.
- A hat lehetőség közül az 1 – 3. és a 1 – 5. hatvány ír le legjobban egy hídkapcsolást, de kicsivel nagyobb hibát kapunk csak, ha az 1. és a 2. hatványt használjuk fel.
- Érdekes jelenség, amit a holtávánál tapasztalunk. Az $\{1, 3, 5\}$ és az 1 – 5. hatvány 18 kHz-ig jobban teljesít a többinél, de ettől fogva az 1. és az 1 – 2. hatvány jobban teljesít.

Általános következtetések:

- Minél nagyobb a mintavételi frekvencia értéke a bemenő gerjesztés felső sávhatárához képest, annál pontosabb lesz a modell. Ha nem megfelelően választjuk meg a hatványok számát, a modell hibajele az átlapolódó komponensek miatt nőhet. Erre megoldást nyújt a bemenő gerjesztés szűrése, bár érdemes megjegyezni, hogy a modell frekvenciafüggő tulajdonságait nem tudjuk pontosan feltérképezni.

- Ha a nemlineáris karakterisztika Taylor-sorba fejthető, a Taylor-sorban nem szereplő együtt-hatók nem javítanak a modellen, de a számítási teljesítményt jelentősen megnövelhetik. Ezért identifikáció során érdemes lehet egy-egy hatványt kihagyni.
- Nincsen olyan hatvány-kombináció, ami általánosan, minden karakterisztikánál a legalacsonyabb hibát adná, de az 1 – 5. hatvánnyal érdemes elkezdni az identifikációt, és utána kihagyni egyet-egyet, hátha egy hatványra nincsen szükség.

A kísérletet LMS algoritmussal is elvégeztem. Az eredményt – ami nagyon hasonló az előbbihez – az A. mellékletben helyeztem el.

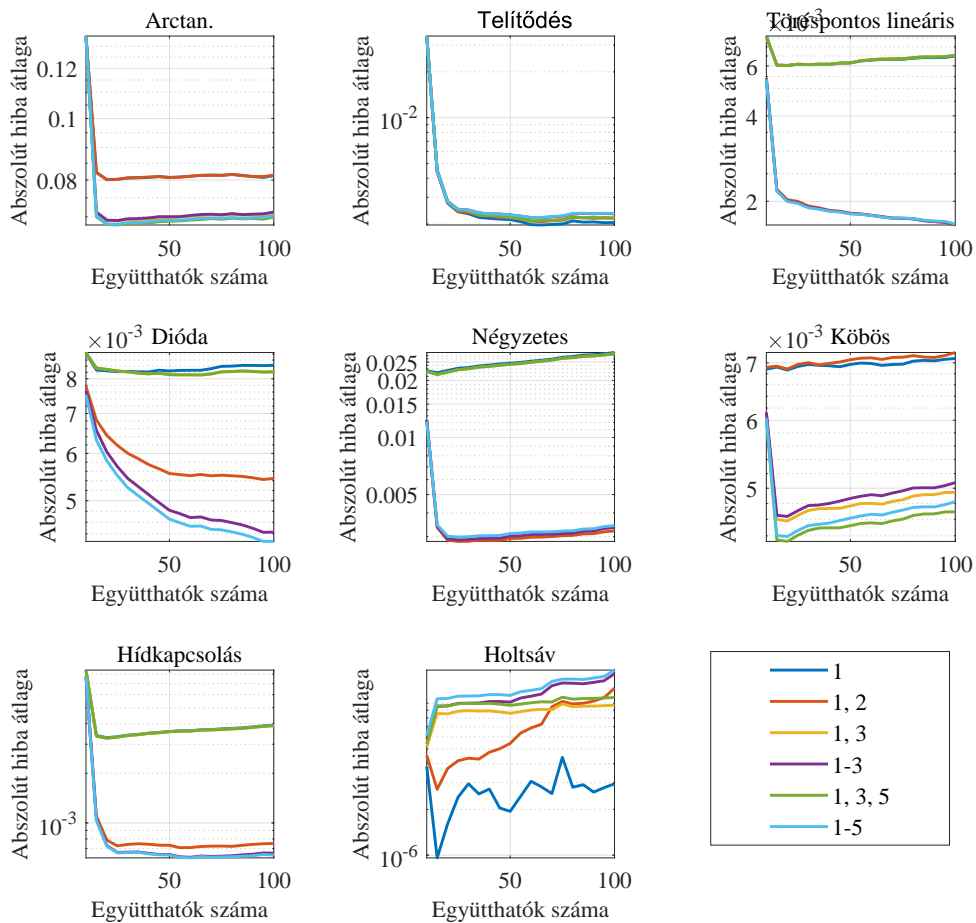
6.2. Együtthatók számának megválasztása

Megvizsgáltam, hogyan hat a Volterra-sor együtthatóinak száma az átlagos abszolút hibára. Az eredményeket a 33. ábra foglalja össze. A szimulációt RLS és LMS algoritmussal is elvégeztem. Hasznosabbnak láttam az LMS ábráját közölni itt, az RLS algoritmusét a mellékletbe tettem.

A szimuláció feltételei megegyeznek az előzővel, annyi különbséggel, hogy a mintavételi frekvencia rögzített 24 kHz volt, és az LMS algoritmus bátorsági tényezőjét minden egyes hatvány esetén 0.01-re rögzítettem.

Megfigyelések, nemlinearitások szerint:

- Egy küszöbérték felett a hiba már nem csökkent tovább az arctan-szerű telítődéses karakterisztikánál. A korábbiakkal összehangban, minél több páratlan hatvány szerepel a szűrőben, annál kisebb a maradó hiba.
- A töréspontos telítődéses karakterisztikánál mind a hatféle hatványkombinációnál nagyon közeli hibaértéket kaptam. Érdekes módon a lineáris szűrő teljesített a legjobban.
- Ha két, $x = 0$ pontban megtörő félegyenesből áll a nemlineáris függvény, azt tapasztalhatjuk, hogy a csak páratlan hatványokat tartalmazó modellek rosszabbul teljesítenek azoknál, melyek páros hatványt is felhasználnak. Előbbi esetben a hiba is növekszik az együtthatók számának emelésével.
- Ha diódát modellezünk, a lehető legtöbb hatványt érdemes felhasználni, de ügyelni kell arra, hogy páros hatvány is legyen a modellben. Ha nincsen páros hatvány, akkor egy kezdeti minimum érték után a hiba az együtthatók számának emelésével nő.
- Ha a karakterisztika egy másodfokú parabolával írható le, azok a modellek teljesítenek jól – az előzetes várakozásoknak megfelelően – melyek tartalmazzák a gerjesztés második hatványát. Ha a szűrő hosszabb, mint az impulzusválasz lényegi része, a hiba növekedni kezd.



33. ábra. Volterra-modell hibajelének alakulása különféle hatványok és együtthatószám esetén, LMS algoritmussal

- Köbös függvényénél hasonló jelenséget figyelhetünk meg. Annak a modellnek alacsonyabb a hibája, amelyik tartalmazza a harmadik hatványt. Az impulzusválasz lecsengése után a hiba növekedni kezd az együtthatók számával együtt.
- Hídkapcsolásnál előnyös, ha a gerjesztés második hatványát felhasználhatja a modell.
- Holsáv modellezésére a Volterra-modell korlátozottan alkalmas – a legjobb eredményt az adja, ha csak lineáris szűrőt használunk. A hiba meredeken emelkedik az együtthatók számának növelésével.

Ebből a kísérletből az a fontos következtetés vonható le, hogy a Volterra-modellnél a hiba az együtthatók számával is összefügg LMS algoritmus esetén. A jelenség oka abban keresendő, hogy a rögzített bátorsági tényező miatt a felesleges (impulzusválasz hosszát meghaladó együtthatók) a 0 érték körül ugrálnak, és ezzel hozzájárulnak a modell hibájához. (RLS algoritmusnál ez a bátorsági tényező adaptív állítása miatt nem így van.)

6.3. Bátorsági tényező megválasztása

Ha a jel abszolút értéke kisebb, mint 1, a magasabb rendű hatványok értéke nagyságrendekkel marad el a bemenő jeltől – az LMS algoritmus beállása azonban összefügg a jel teljesítményével (ezt használja ki az NLMS algoritmus).

Megvizsgáltam, hogy a magasabb hatványok bátorsági tényezőjét érdemes-e nagyobb állítani. A szimuláció során mind a nyolc karakterisztikát felhasználtam. A körülmények megegyeztek az előzővel, azzal a különbséggel, hogy az első öt hatvány mindegyikét felhasználtam, és a szűrő hosszát 50-re állítottam. A bátorsági tényezők a következők voltak: [0.0001, 0.0001, 0.0001, 0.0001, 0.0001], [0.0001, 0.0002, 0.0003, 0.0004, 0.0005], [0.0001, 0.0004, 0.0009, 0.0016, 0.0025] és [0.0001, 0.0008, 0.0027, 0.0064, 0.0125].

Az eredményeket – az abszolút hiba mozgóátlagát a lépésszám függvényében – a 34. ábra foglalja össze. Könnyen felismerhető, hogy a holtzávon kívül az összes karakterisztika esetén a [0.0001, 0.0008, 0.0027, 0.0064, 0.0125] bátorsági tényezők vezettek a legjobb eredményre, így megállapítható, hogy a magasabb hatványokhoz érdemes nagyobb bátorsági tényezőt rendelni.

6.4. Kompenzálás a Volterra-moddal

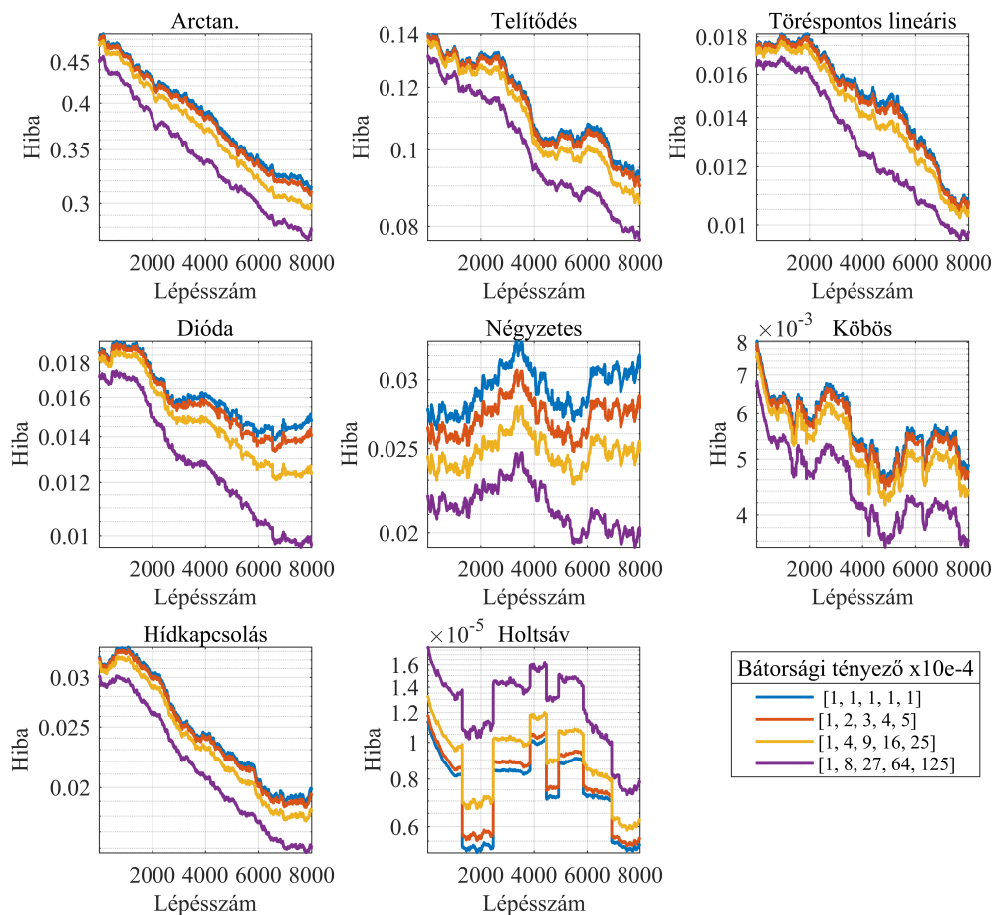
Ebben a részben elsőként azt mutatom meg, hogy a Volterra-modell felhasználható egy dinamikus nemlineáris rendszert online kompenzálására, majd egy gyakorlati példán – egy shaker-en – demonstrálok a működését. A kompenzálás lényegét a 3. ábra mutatja be.

Tegyük fel, hogy adott egy Hammerstein-moddal leírható rendszer, melynek lineáris dinamikus szűrőjét a (41) jellemzi, nemlineáris karakterisztikáját pedig az $\arctan(5x)$ függvény. Célunk az, hogy úgy kompenzáljuk, hogy a gerjesztés és a (fizikai) kimenet között a lehető legkisebb legyen a torzítás.

Mivel a hibajel visszacsatolása dinamikát tartalmaz, azt nem csatolhatjuk közvetlenül vissza az adaptáció során anélkül, hogy nagy valószínűséggel instabillá tennénk a rendszert. Erre nyújt megoldást az ELMS és az XLMS algoritmus, melyeket korábban már bemutattam az 5.4.1. és az 5.4.2. alfejezetben.

Ha a rendszernek fizikai kimenete van, a kettő közül csak az XLMS algoritmust használhatjuk, mivel az ELMS igényli a rendszer offline kialakított inverzét, de ezt stabilan csak úgy határozhatjuk meg, ha az inverzet a rendszer mögé kötjük az adaptáció során. Az XLMS azonban a rendszer offline identifikált modelljét használja csak fel. Az XLMS algoritmusnál a kívánt kimeneti jel a késleltetett bemenő gerjesztés lesz. Késleltetésnek 30 lépésnyit választottam. Az impulzusválasz lecsengésénél nem sokkal hosszabbra vettem a modellt (50) és a szabályzó (80) együtthatóinak számát. Előbbihez az első három, utóbbihoz az első öt hatványt használtam, mivel [15] szerint a modellnek nem kell pontosnak lennie.

A gerjesztés 24 kHz-es egyenletes eloszlású zaj volt, melyet egy 1.5 kHz felső sávhatáru aluláteresztő szűrővel szűrtem a MATLAB DSP Toolbox-ának segítségével. Az aluláteresztő



34. ábra. A bátorsági tényező hatása a konvergencia sebességére Volterra-modellnél.

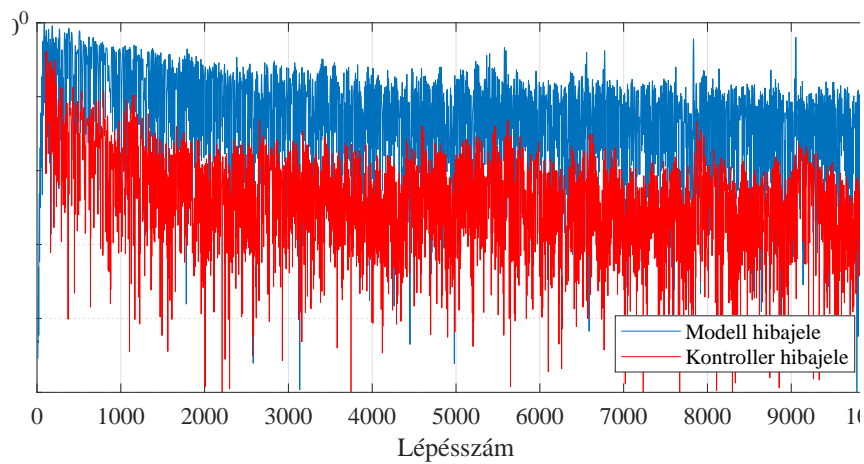
szűrő fokszáma 100 volt, és 90 dB-es elnyomással rendelkezett a zárótartományon.

A modellt 10000 lépésen keresztül offline identifikáltam, majd a következő 10000 lépésre bekötöttem az XLMS algoritmus struktúrájába. A controller és a modell abszolút hibajelét a 35. ábrán mutatom be. Jól látszik, hogy a modell hibája magasabb, mint a controlleré.

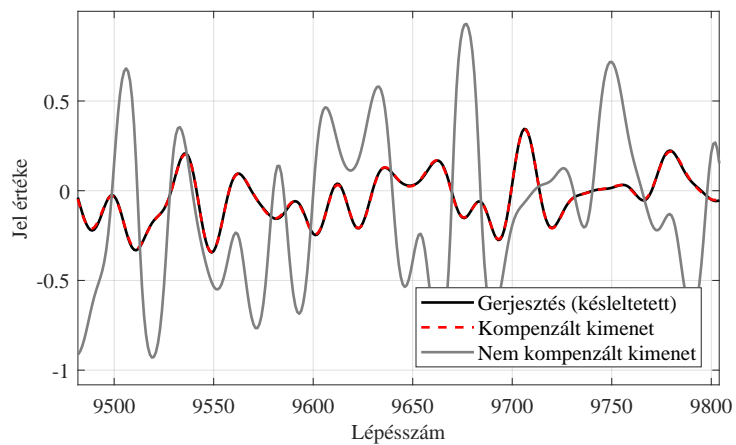
Ennek ellenére a controller úgy képes kompenzálni a rendszert, hogy kimeneti jele alig különbözik a késleltetett gerjesztéstől. A kompenzáció sikerességét a koherenciafüggvény is alátámasztja (37. ábra).

6.4.1. Egy shaker modellezése és kompenzálása

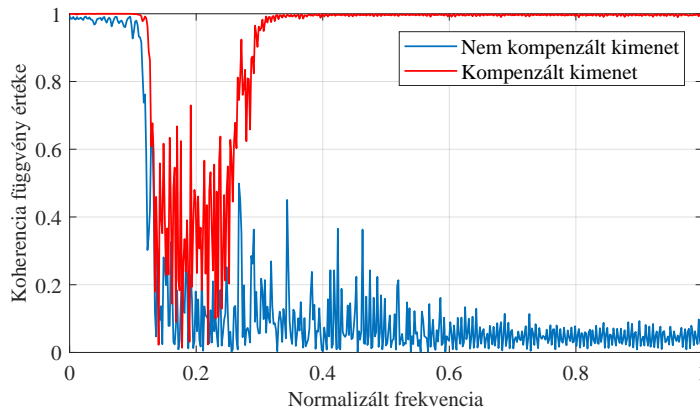
Egy valós példán is bemutatom, hogy a Volterra-modell alkalmas egy nemlineáris dinamikus rendszer kompenzálására. Ehhez egy BRÜEL & KJÆR 4810 Mini-Shaker-t eszközt használok fel. Ahogyan említettem, a BRÜEL & KJÆR 4810 Mini-Shaker jellemzője, hogy alacsony frekvenciás (30 Hz körüli) gerjesztésnél nemlineáris viselkedést mutat (38. ábra) – ez 300 Hz környékén már



35. ábra. XLMS algoritmus offline modelljének és az online adaptált kontrollerének hibajele.

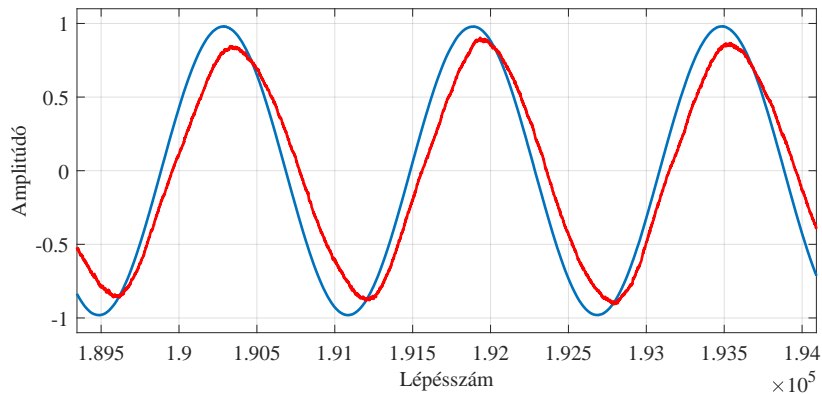


36. ábra. Késleltetett gerjesztés, Volterra-moddal kompenzált és kompenzálatlan kimenet arctan karakterisztikánál, 1.5 kHz-es felső sávhatárú zaj gerjesztésnél.



37. ábra. Volterra-moddal kompenzált és kompenzálatlan rendszer kimenetének koherenciafüggvénye arctan-szerű karakterisztikánál és szűrt bemenő zaj gerjesztésnél.

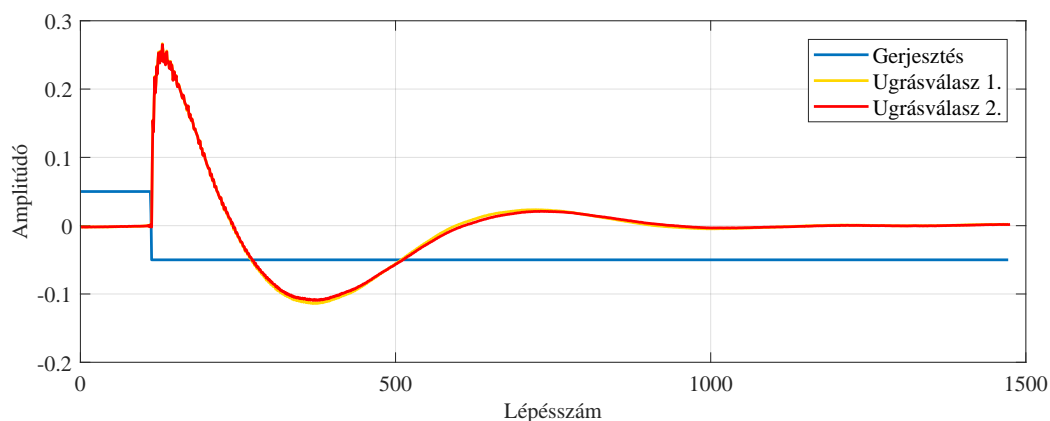
eltűnik, és inentől kezdve lineáris rendszerként tekinthetünk rá. 30 Hz-es szinuszos bemenő jelnél a kimeneti jelének torzítása: $\text{THD} = 1.7\%$.



38. ábra. Shaker gerjesztése és kimeneti jele.

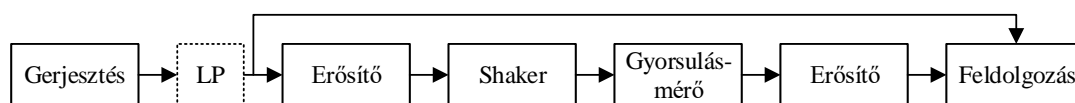
Azért, hogy többféle megközelítést is hatékonyan ki tudjak próbálni, elsőként különféle gerjesztésekre rögzítettem a shaker kimeneti jelét, majd MATLAB-ban megalkottam a shaker modellt 1-3. hatványt és 1000 együtthatót felhasználva. Azért 1000 együtthatót használtam fel, mert úgy tűnt, hogy a shaker ugrásválasza 1000 együttható után már lecseng. Ezt mutatja a 39. ábra, ahol két egymás utáni ugrásválaszt is megfigyelhetünk – ezek némileg eltérnek egymástól, feltehetően a mérés nem ideális körülményei miatt (többek között a shaker-hez rögzített gyorsulásmérő kábele befolyásolhatta a mérést).

A modell megalkotásához az 40. ábrán látható elrendezést használtam. A függvénygenerátor



39. ábra. A shaker ugrásválasza

jele egy erősítőn (West Sound DS 150) halad keresztül, majd a shaker-re kerül. A shaker kimeneti jelét egy gyorsulásmérő szenzor alakítja elektromos jellé, melyet erősíteni kell (BRÜEL & KJÆR Nexus Conditioning Amplifier), és ezt követően lehet feldolgozni. Zaj gerjesztésnél a zajgenerátor (Hewlett Packard HO1-3722A Noise Generator) mögé kell kapcsolni egy aluláteresztő szűrőt (Krohn-Hite model 3322 filter) is.

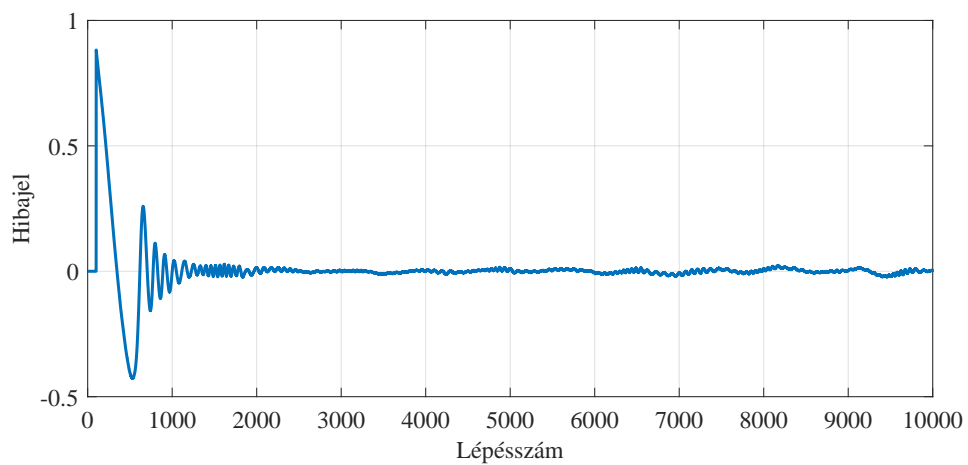


40. ábra. Elrendezés shaker jeleinek méréséhez.

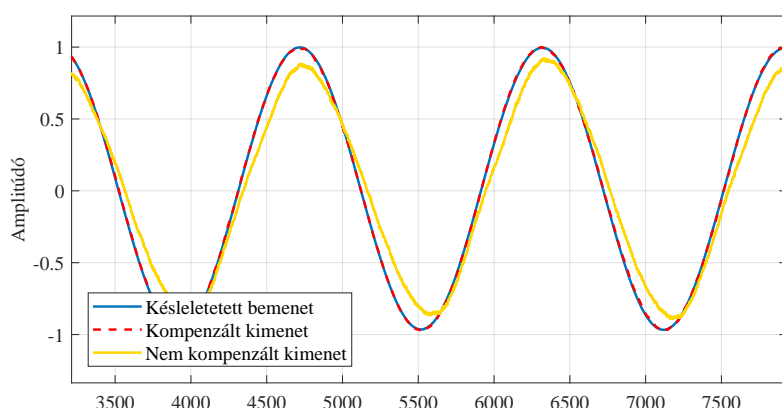
Az előzőhöz hasonló módon XLMS algoritmussal linearizáltam a shaker modelljét – nem az eredeti eszközt, de így többféle lehetőséget is gyorsan kipróbálhattam. A szabályzó és a modell együtthatóinak száma is 1000-1000 volt. A szabályzóhoz az első öt, a modellhez az első három hatványt használtam. Az XLMS algoritmus hibajelét a 41. ábra mutatja.

A 42. ábra mutatja a shaker késleltetett bemenő jelét, illetve a kompenzálatlan és a kompenzált kimenetet. Jól látható, hogy az XLMS algoritmus képes linearizálni a rendszert. A kimenő jelre torzítatlan esetben $THD = 0.815$, míg kompenzáció után $THD = 0.079$ volt.

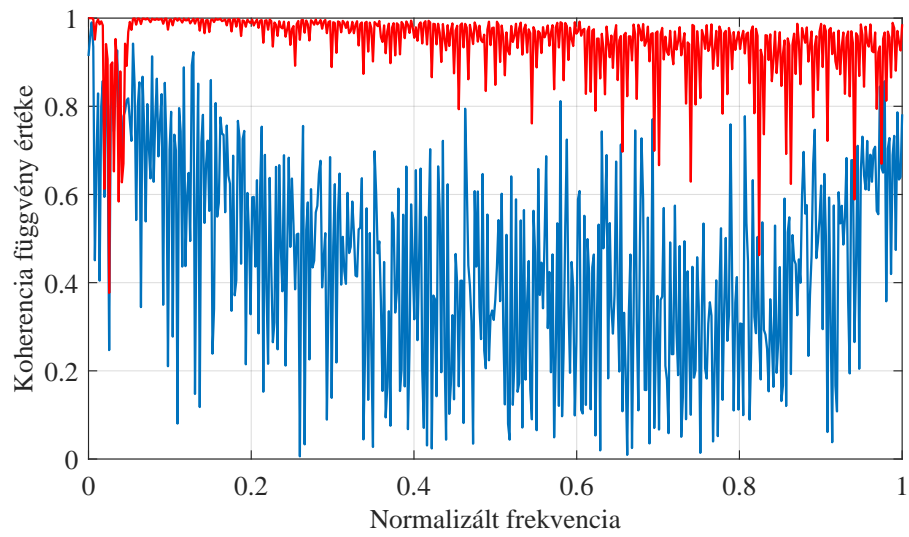
A be és kimeneti jel között szemmel nem lehet különbséget tenni. Az eltérés bemutatására ezért kirajoltam a koherenciafüggvényt is (43. ábra). Jól látható, hogy a kompenzált rendszer torzítása csökkent.



41. ábra. Kontroller hibajele shaker kompenzálásakor.



42. ábra. Shaker késleltetett gerjesztése, kompenzált és nem kompenzált kimenete.



43. ábra. A shaker Volterra-modellel kompenzált és nem kompenzált kimenetének koherenciafüggvénye

7. Gyakorlati tapasztalatok a Hammerstein-moddellel

Ebben a részben a Hammerstein-moddellel kapcsolatos gyakorlati tapasztalataimat összegzem. Az a szakirodalom, amit feldolgoztam, ezeket az alapvető vizsgálatokat nem írja le annak ellenére, hogy feltehetően felmerültek munka közben. Ezt a hiányosságot szeretném most pótolni.

A Hammerstein-moddellel történő identifikációhoz a 25. ábrán látható struktúrát használtam fel, mely biztosítja, hogy a bemenő gerjesztés és egyes értékeihez tartozó hiba megfelelően álljon párba az adaptáció során. Megállapításaim kizárólag erre a struktúrára érvényesek, így elképzelhető, hogy például maximum likelihood módszerrel [49–51] más eredményre jutnánk.

A Hammerstein-moddell egyszerűen használható eszköz egy nemlineáris rendszert kompenzálására, mivel a nemlineáris blokk invertálása elegendő a szabályzóhoz.

7.1. Modellezés a Hammerstein-moddellel

A Hammerstein-moddell a spline függvénynek köszönhetően képes olyan nemlineáris karakterisztikákat is pontosan modellezni, melyek egy Volterra-sor alapú modellnek problémát jelentenek.

7.1.1. Hammerstein-moddell alapú rendszer identifikálása Hammerstein-moddellel

A 44. és a 45. ábra a (41) dinamikus szűrőt és egy diódát, illetve egy holsávot tartalmazó Hammerstein-moddell alapú rendszer identifikációjának eredményét mutatja. Az ábrák elkészítéséhez 200 ezer lépést, 24 kHz-es mintavételi frekvenciát, szűretlen egyenletes eloszlású, -1 és 1 közötti gerjesztést használtam. A szűrő együtthatóinak száma 100, az inverz hossza 200 volt. A karakterisztika 21 kontrollpontot tartalmazott. Az együtthatókat LMS algoritmussal tanítottam. A FIR szűrő bátorsági tényezője 0.01, a kontrollpontoké 0.001. Azért, hogy egyetlen megoldás létezzen, a FIR szűrő együtthatóinak maximumát 1-re állítottam be, ami működőképes megoldásnak bizonyult.

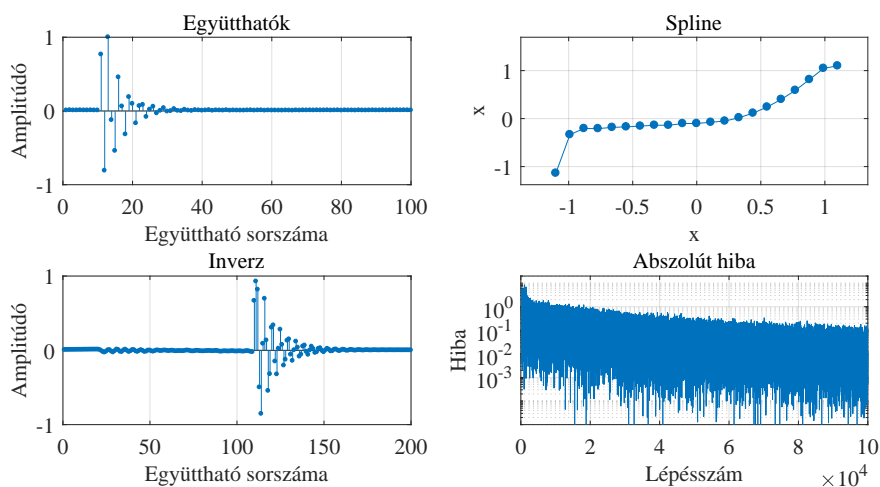
Jól látható, hogy mindkét esetben mind az együtthatók, mind az inverz együtthatói lecsengenek, így nem indokolt hosszabb inverzet használni. A hiba folyamatosan csökken.

Érdemes kiemelni, hogy ennél a blokk alapú modellenél bármelyik blokkban lehet erősítés, melyet a másik kompenzál.

A dióda identifikálásánál egyszerre tanítottam a FIR szűrőt és a spline függvényt, míg a holsávnál az első ezer lépésben csak a FIR szűrőt tanítottam, majd egyszerre a kettőt.

A tanítás ilyen módon történő szétválasztását az indokolhatja, hogy a FIR szűrőnek legyen ideje megközelítőleg beállítani a késleltetést, így a bekapcsolódó spline függvény már megfelelően párosított hibajeleket és bemenő gerjesztést kap. A hibajel megugrását megfigyelhetjük a 45. ábrán, de utána ez jelentősen, a lineáris szűrő hibajele alá csökken.

Tapasztalataim szerint nem kell feltétlenül késleltetni a nemlineáris blokk adaptációját, mert



44. ábra. Diódát tartalmazó rendszer identifikálása
Hammerstein-moddal

enélkül is megfelelően, és azonos mértékben csökken a hiba.¹⁰ A dióda karakterisztikánál a 44. ábrán ezt mutatom be.

7.1.2. Wiener-modell alapú rendszer identifikálása Hammerstein-moddal

Az előző részben bemutattam, hogy a Hammerstein-modell képes Hammerstein-moddal leírható nemlineáris dinamikus rendszereket identifikálni. Jogosan merül fel a kérdés, hogy Wiener-modell alapú rendszereknél is képes-e ilyenre.

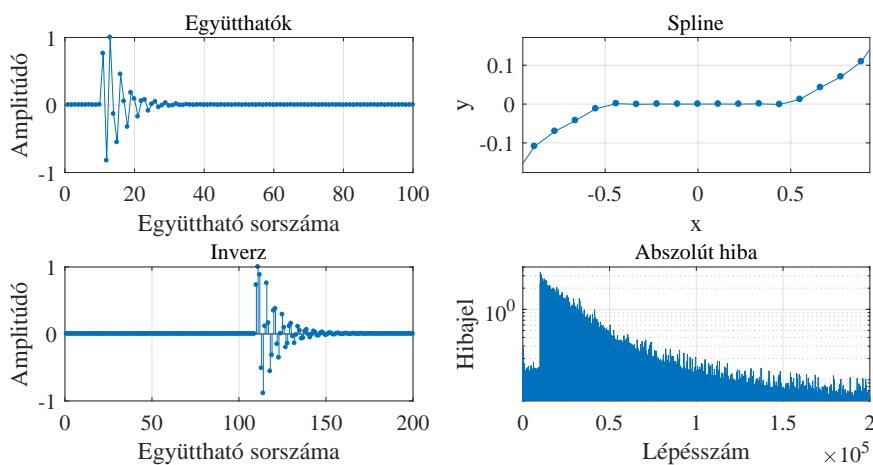
A szimuláció során az előbbivel mindenben megegyező környezetben vizsgáltam a kérdést, két dolgot kivéve: a Hammerstein-modell helyett a statikus nemlineáris blokkot a lineáris dinamikus szűrő mögé kötöttem, és megnöveltem a késleltetett inverz együtthatóinak számát 1000-re, mivel kezdeti kísérleteim során a lecsengéshez ennyi együtthatóra volt szükségem. Az inverz alakja azonban kísérletről kísérletre változott.

Az 46. ábra jól mutatja, hogy az a struktúra, amit az adaptációhoz használtam nem képes a Wiener-modellt identifikálni.

Megállapítottam továbbá, hogy:

- Az eredményre nincsen hatással az, hány lépésig tanítom a lineáris blokkot, mielőtt a spline adaptálását is elkezdeném.
- Nem javít az eredményen, ha a kezdeti néhány lépésben, a spline adaptálása előtt csökkentett (az eredeti amplitúdó 1 százalékának megfelelő) gerjesztést vezetek a bemenetre. Ezt az indokolhatná, hogy ilyen kis tartományon a karakterisztika lineárisnak tekinthető.

¹⁰Ez a megállapítás arra az adaptációs struktúrára és azokra a karakterisztikákra érvényes, melyeket felsoroltam.



45. ábra. Holtsávot tartalmazó Hammerstein-modell alapú rendszer identifikálása Hammerstein-moddellel

Ez az eredmény arra világít rá, a gyakorlati életben körültekintően kell eljárunk, amikor Hammerstein-modellt használunk fel.

7.1.3. Shaker modellezése Hammerstein-moddellel

Hammerstein-moddellel identifikáltam a korábban bemutatott shaker-t is. Ehhez felhasználtam a shaker korábban rögzített kimeneti jelét és gerjesztését. A mintavételi frekvencia 48 kHz volt, miközben a shaker bemenetére olyan fehér zajt kötöttem, melyet előzetesen egy 50 Hz-es aluláteresztő szűrővel szűrtem. A mérés elrendezését a 40. ábrán láthatjuk.

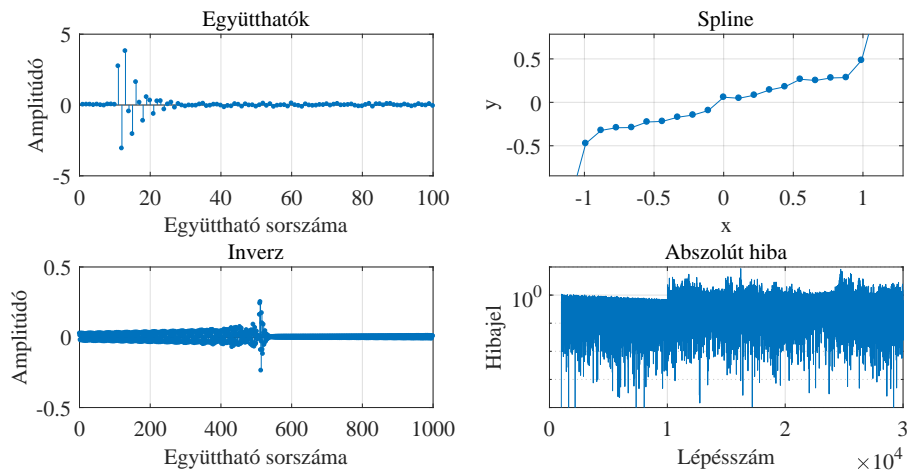
A shaker identifikálására a 25. ábrán bemutatott struktúrát használtam. A 39. ábrán jól látszik, hogy a shaker ugrásválasza 1000 lépés után cseng le, így a FIR blokk együtthatóinak számát 1500-ra állítottam. A spline-függvény 31 kontrollponttal rendelkezett.

Az első 10 ezer lépésben csak a lineáris blokkot, majd az azt követő 40 ezer lépésben csak a nemlineáris blokkot tanítottam, végül 150 ezer lépésben már együtt a kettőt.

A 47. ábrán azt láthatjuk, hogyan alakult az identifikáció visszacsatolt hibajele, a lineáris blokk impulzusválasza, illetve a nemlineáris blokk kontrollpontjainak értéke.

Az impulzusválasz nagyon zajos, és a hibajel sem csökken kívánatos mértékűre (0.01-környékén állandósul). Láthatjuk, hogy a nemlineáris blokk szerint a shaker enyhén nemlineáris, elsősorban a két végső állásban tér el az ideális egyenestől. Ez összhangban áll a nemlineáris függvény közelítő meghatározásakor kapottakkal (19) és azzal, hogy a shaker mozgórészét visszahúzó membrán nagy kitéréseknél mutat nemlineáris viselkedést.

Összességében kijelenthető, hogy a 25. ábrán bemutatott struktúra és a Hammerstein-modell korlátozottan alkalmas a shaker identifikálásra. Erre az lehet a magyarázat, hogy a shaker inkább Wiener-modell szerint működik, amit az általam felhasznált eljárás nem identifikál megfelelően.



46. ábra. Wiener-modell alapú, arctan függvényt tartalmazó rendszer Hammerstein-moddellel történő identifikálásának eredménye.

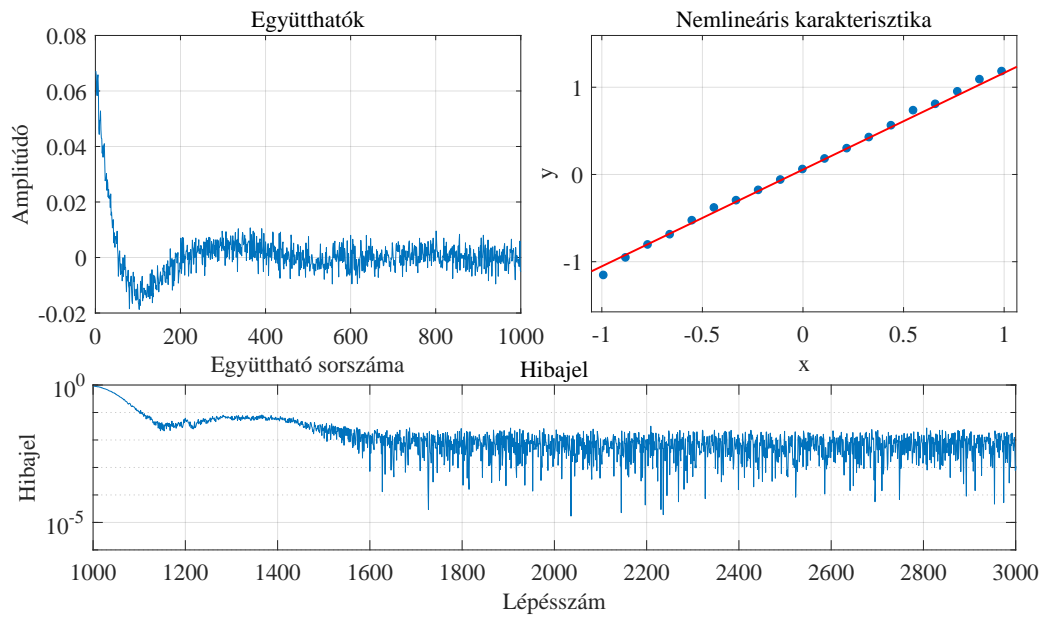
7.2. A kontrollpontok és az együtthatók számának megválasztása

Felmerül a kérdés, hogyan válasszuk meg a lineáris blokk együtthatóinak és a spline-függvény kontrollpontjainak számát. A spline függvény rugalmassága miatt bármilyen gyengén nemlineáris függvényt képes modellezni, így a nyolc karakterisztika közül egyet választottam a kérdés alaposabb vizsgálatához: az arctan. függvényt. A 48. ábra azt mutatja, mekkora az átlagos abszolút hiba értéke az 50000. és a 60000. lépés között. Ennyi lépés után tapasztalatom szerint beállt a rendszer. Az inverz szűrő hossza minden esetben 300 volt, miközben a kontrollpontok és a szűrő együtthatóinak számát változtattam. LMS algoritmussal adaptáltam a modellt. A FIR bátorsági tényezője 0.01, a kontrollpontoké 0.01 volt.

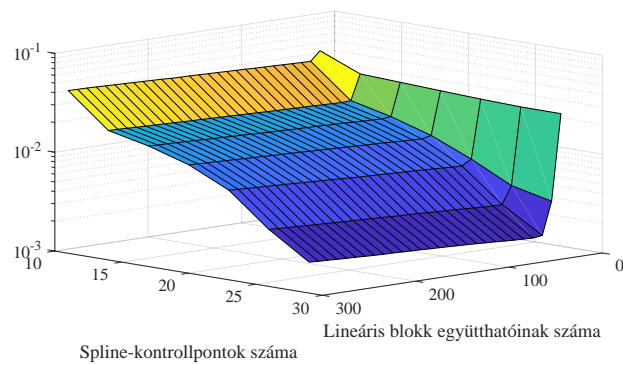
Az ábráról kiderül, hogy ha a több spline pontunk van, jobb eredményt kapunk, és az is jól látszik, hogy az együtthatók számának növelése nem befolyásolja számottevően az eredményt, miután a FIR szűrő hossza meghaladja a rendszer impulzusválaszának lényegi részét. Az LMS algoritmus fix bátorsági tényezője miatt azonban az együtthatók számának szükségesnél tovább történő növelése magasabb hibához vezet.

7.3. A bemenő jel szűrése

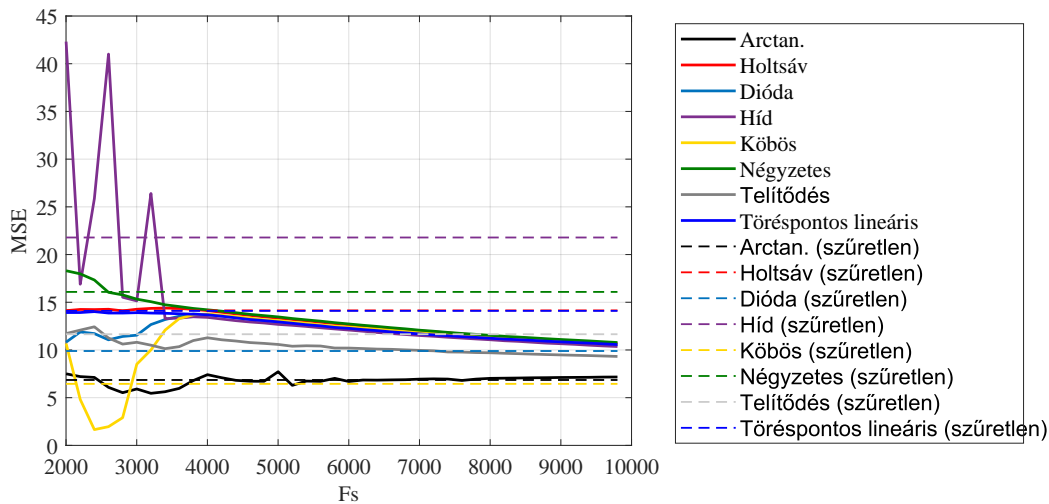
A Hammerstein-modellnél is megvizsgáltam, hogyan hat a bemenő gerjesztés szűrése a modell abszolút hibájára. A 49. ábrát úgy készítettem, hogy a bemenő jelet állandó 1 kHz felső sávhatárú aluláteresztő szűrővel szűrtem, és a mintavételi frekvenciát növeltem. A többi paraméter megegyezett az előző kísérlettel – a különbség mindössze annyi volt, hogy nem a maradó hiba abszolút értékének az átlagát vettem, hanem az eredeti és az identifikált statikus nemlinearitás közötti négyzetes különbséget.



47. ábra. Shaker-identifikálása Hammerstein-moddellel.



48. ábra. Az együtthatók és a kontrollpontok számának hatása a Hammerstein-modell hibajelére



49. ábra. Bemenő jel szűrésének hatása a Hammerstein-modell teljesítményére

Az ábrán szaggatott vonallal jelöltem azt az esetet, amikor szűrés nélkül vezettem 10 kHz mintavételi frekvenciájú zajt a bemenetre. Kitűnik, hogy bár általánosan hatékonyabbá teszi a Hammerstein-modellt a szűrés, több karakterisztika esetén is – mint amilyen a köbös és a dióda – a szűretlen gerjesztés teljesít a legjobban.

Ezt a jelenséget az magyarázhatja, hogy a FIR szűrőnek és az inverzének kialakításához a szűretlen zaj-gerjesztés a legjobb környezet.

7.4. Offline kompenzálás a Hammerstein-moddellel

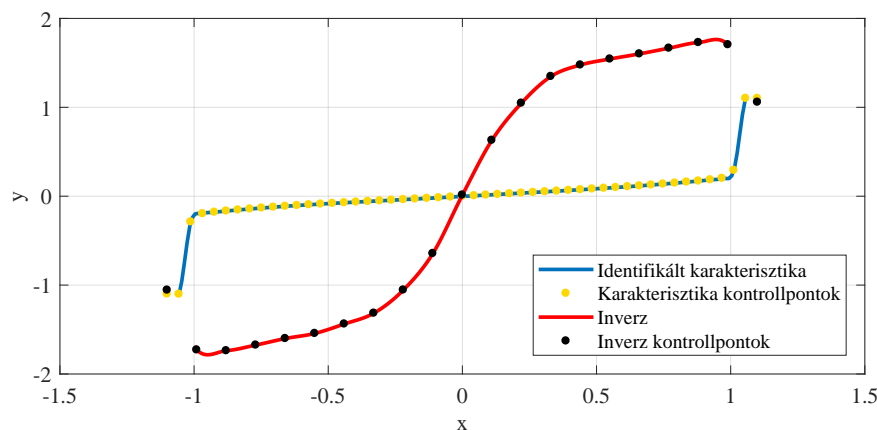
Ha egy nemlineáris dinamikus rendszert az 25. ábrán látható struktúrával identifikáltunk, könnyen készíthetünk olyan szabályzót (offline), melyet a rendszer elé kötve kompenzálhatjuk a gerjesztés és a kimenet közötti torzítást.

A szabályzónak két eleme van, a korábban már bemutatott 3. ábra szerint. Szükségünk van a lineáris blokk és a nemlinearitás inverzére. Előbbi a 25. ábrán látható struktúrával az identifikáció során már előállt. A nemlineáris blokk inverzét amennyiben az invertálható adaptív módon állíthatjuk elő, például a 28. ábra szerinti eljárással.

A kompenzáció szemléltetéséhez a (41) rendszert használtam, melynek bemenete elé az $\arctan(5x)$ függvényt kötöttem (Hammerstein-modell). A bemenő, -1 és 1 közötti egyenletes eloszlású zaj gerjesztés mintavételi frekvenciája 24 kHz, a lépésszám 200 ezer volt. A spline kontrollpontokat és a FIR szűrőt LMS algoritmussal tanítottam. A kontrollpontok bátorsági tényezője 0.001, és 0.01 a FIR szűrőé. 100 együtthatót használtam a szűrőhöz, 200-at az inverzéhez.

A spline függvényhez 21, az inverzéhez 51 kontrollpontot használtam fel. Az inverzet offline, adaptív módszerrel tanítottam a 3. ábrának megfelelően. A tanításhoz szinuszjelet használtam

azért, hogy minden kontrollpont elegendő gerjesztés jusson. Eközben azt figyeltem meg, hogy a karakterisztika inverzének alsó és felső tartománya nem állt be tökéletesen. Ennek oka az, hogy a spline függvény egy-egy intervallumbeli viselkedésére négy kontrollpontnak van hatása, de a két szélén elhelyezkedő utolsó egy-egy kontrollpontot csak „egyoldalúan” taníthatjuk. Így amikor a rendszer kompenzáljuk, a maximálisnál érdemes alacsonyabb amplitúdójú gerjesztést alkalmazni – ezzel elkerülhető, hogy a legszélső intervallumról származzon érték. A kontrollpontok számának megfelelően nagyra állításával ez a hatás minimálisra csökkenthető.

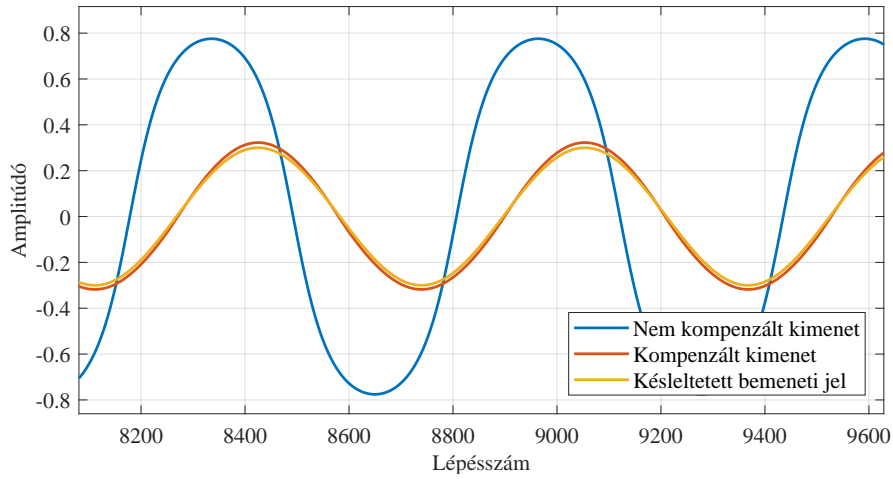


50. ábra. Identifikált spline függvény és inverze.

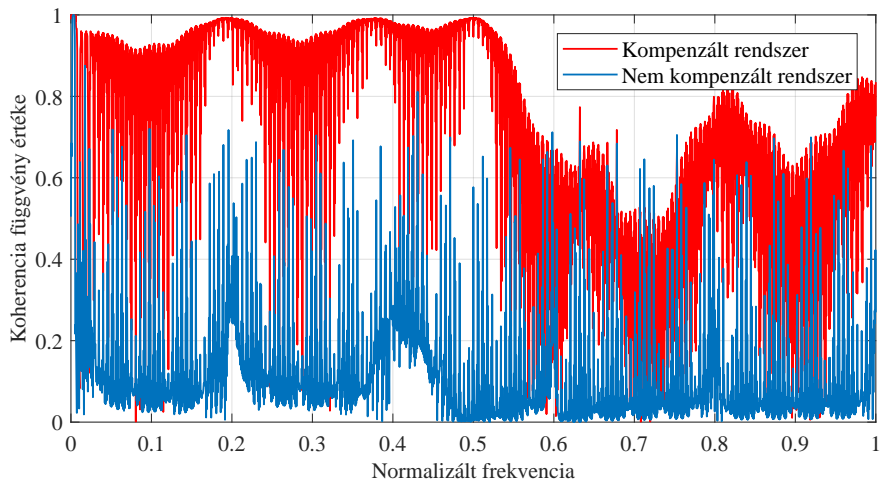
Az identifikált karakterisztika és az offline adaptív módon kialakított inverzét az 50. ábra mutatja. A karakterisztika meredeken emelkedik az $x = 0$ körüli tartományon, ezért az inverz laposabb. Az ábrán nem feltétlenül látszik, de a függvény és inverzének kimenete az $x = y$ egyenest adja ki.

A korábban bemutatott adaptációs struktúra a modellel együtt az inverzet is kialakítja. Így ennek és az inverz lineáris bloknak a sorbakötésével tudjuk kompenzálni a rendszert.

A 51. ábra szinuszjel gerjesztésnél mutatja a rendszer eredeti és kompenzált kimenetét, illetve a késleltetett bemenő gerjesztést. Szemmel láthatóan a bemenet és a kimenet kompenzálás után is eltér egymástól, de a torzítás jelentősen lecsökkent, amit az 52. ábrán látható két koherencia függvény is jól mutat. A függvényeket a MATLAB *mcoherce* nevű függvényével generáltam.



51. ábra. Hammerstein-moddal kompenzált és nem kompenzált kimenet Hammerstein-alapú rendszer és arctan karakterisztika esetén.



52. ábra. Hammerstein-moddal kompenzált arctan alapú rendszer kompenzált és nem kompenzált kimeneti jelének koherenciafüggvénye

8. Összefoglalás, kitekintés

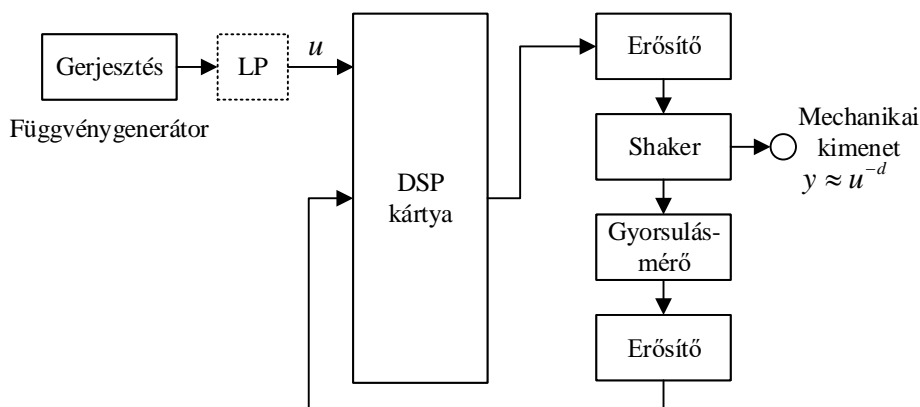
A legtöbb gyakorlati alkalmazás során kívánatos, hogy az a rendszer, amivel dolgozunk, lineáris legyen, mivel a lineáris rendszerek kezelésének eszköztára jól kidolgozott, és általánosan ismert. A nemlineáris rendszereket ezért kompenzálni kell. Ezt a gyakorlati igényt tartottam szem előtt a dolgozat témájának feldolgozása során. A lineáris kapcsolatot, vagyis az egységnyi erősítést úgy lehet elérni, hogy a szakasz elé vagy mögé sorosan kapcsolunk egy olyan szabályzót, ami megegyezik a szakasz (késleltetett) inverzével. Ha a szakasz mögé tesszük, nem leszünk képesek fizikai (akusztikus, mechanikai) kimenetű rendszereket linearizálni, így a dolgozatban a másik lehetőséget vizsgáltam.

A feladat sokszínűségéből fakadóan a kompenzálásra – a kimenet és a bemenet közötti torzítás minimalizálása – számos eszköz létezik. Ezek eltérnek abban, hogy az inverzet online vagy offline állítják elő. Előbbi folyamatosan képes egy lassan változó rendszerhez igazodni, míg az utóbbi nem, a rendszer megváltozásakor újra el kell végezni a műveletet. Az inverz online meghatározásánál az jelent nehézséget, hogy a közvetlenül visszacsatolt hibajel az adaptációs hurok dinamikus tulajdonságai miatt instabillá teheti a struktúrát. Az inverz offline meghatározásakor pedig készülnünk kell arra, hogy egy nem minimálfázisú rendszer (vagyis amikor az átviteli függvényben pólusok vannak az egységkörön kívül) inverze instabil lesz. Ebben az esetben a késleltetett (közelítő) inverzet kell meghatározni, mely minden esetben stabil lesz.

A dolgozatban bemutattam a nemlineáris rendszerek kétféle csoportosítását, és részletesebben vizsgáltam két modellt, a Volterra- és a Hammerstein-modellt. Előbbi a Volterra-sorra épül, és egyszerűsített esetben a bemenő jel hatványait használja a kimeneti jel kiszámítására. A Hammerstein-modell egy olyan blokkalapú modell, mely egyetlen (sorbakapcsolt) statikus nemlinearitást, és egy lineáris dinamikus szűrőt tartalmaz. Mindkét modell paramétereiben lineáris modell, így gradiens és ahhoz hasonló adaptív algoritmusokkal (LMS, RLS) taníthatóak.

A Volterra-modellnél bemutattam, hogy a gerjesztés hatványai miatt, ha nem megfelelően nagy a mintavételi frekvencia, nem tapasztalunk optimális működést az átlapolódó komponensek hatására. Megvizsgáltam az együtthatók és a hatványok számának hatását is nyolcféle karakterisztika modellezésénél LMS és RLS algoritmusra. Azért, hogy bizonyítsam, a Volterra-modell képes egy nemlineáris dinamikus rendszert kompenzálni, egy arctan alapú, és egy valós rendszert – egy shakert – is linearizáltam. A shaker alacsony frekvenciás jelek esetén nemlineáris, ami jól illeszkedett a Volterra-modell tulajdonságaihoz. A kompenzálás sikerességét a gerjesztés és a kompenzált, illetve nem kompenzált kimeneti jel közötti koherencia függvényvel bizonyítottam.

A Hammerstein-modellnél a memóriamentes nemlinearitás modellezésére számos lehetőség közül választhatunk (polinom, szakadó polinom, szakaszonként lineáris függvények). Én Catmull-Rom spline-okat használtam erre a célra, mivel rendkívül rugalmasan képesek nemlineáris karakterisztikákat leírni. A Hammerstein-modellnél megvizsgáltam a mintavételi frekvencia, a FIR szűrő és a spline kontrollpontok számának hatását az eredményre. Kiderült, hogy a bemenő



53. ábra. Shaker DSP kártyával történő kompenzálása

gerjesztés szűrése ronthatja a modell teljesítményét. A szűrő fokszáma addig javít a modellen, amíg el nem éri a rendszer impulzusválaszának lényegi hosszát, azt követően az LMS algoritmus rögzített bátorsági tényezőjének következtében növeli a hibát. A spline kontrollpontok számának növelése azonban minden esetben a hiba csökkenéséhez vezet. Mivel a Hammerstein-modell nem nyújt jó teljesítményt, ha a gerjesztést aluláteresztő szűrővel szűrjük, az alacsony frekvencián nemlineáris shaker-t nem voltam képes linearizálni vele. De egy arctan. alapú rendszeren sikeresen demonstráltam, a Hammerstein-modell is alkalmas bizonyos körülmények között a kompenzációra.

A jövőben érdemes lehet többféle karakterisztikát is megvizsgálni, és a Hammerstein-modell visszacsatolással alkalmassá tenni egy dinamikus rendszer viselkedésének pontosabb modellezésére. A következő időszakban a kompenzálásra alkalmas rendszert DSP kártyán is megvalósítom. A tervezett elrendezést az 53. ábra mutatja. A kártya egyik bemenete egy függvény, vagy zajgenerátorhoz fog kapcsolódni. A másik bemenete a shaker kimeneti jele lesz. A DSP kártya hajtja meg a nemlineáris rendszert az erősítőn keresztül úgy, hogy a shaker kimeneti jele a lehető legkevésbé térjen el a bemeneti jeltől. Az elrendezés nagy előnye, hogy a jelek fizikai átkapcsolása nélkül képes a rendszer identifikációját és kompenzálását is elvégezni. A DSP kártyán a Volterra- és a Hammerstein-modell is megvalósítom majd, miközben figyelek a DSP kártya elérhető számítási kapacitásának korlátaira.

Hivatkozások

- [1] G. Liu, F. Wang, F. Qian, and X. Xu, „Identification of Hammerstein systems using key-term separation principle, auxiliary model and improved particle swarm optimisation algorithm,” *IET Signal Processing*, vol. 7, no. 8, pp. 766–773, 2013. [Online]. Available: <http://digital-library.theiet.org/content/journals/10.1049/iet-spr.2013.0042>
- [2] M. Gasparini, L. Romoli, S. Cecchi, and F. Piazza, „Identification of Hammerstein model using Cubic Splines and FIR filtering,” in *International Symposium on Image and Signal Processing and Analysis*, 2013, pp. 347–352.
- [3] J. Sjöberg and J. Schoukens, „Initializing Wiener – Hammerstein models based on partitioning of the best linear approximation,” *Automatica*, vol. 48, no. 2, pp. 353–359, 2012. [Online]. Available: <http://dx.doi.org/10.1016/j.automatica.2011.07.007>
- [4] M. Scarpiniti, D. Comminiello, R. Parisi, and A. Uncini, „Nonlinear spline adaptive filtering,” *Signal Processing*, vol. 93, no. 4, pp. 772–783, 2013. [Online]. Available: <http://dx.doi.org/10.1016/j.sigpro.2012.09.021>
- [5] V. J. Mathews, „Adaptive Polynomial Filters,” *IEEE Signal Processing Magazine*, vol. 8, no. 3, pp. 10–26, 1991.
- [6] H.-W. C. H.-W. Chen, „Modeling and identification of parallel nonlinear systems: structural classification and parameter estimation methods,” in *Proceedings of the IEEE*, vol. 83, no. 1, 1995, pp. 39–66. [Online]. Available: <http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=362753>
- [7] A. Bolstad, B. A. Miller, J. Goodman, J. Vian, and J. Kalyanam, „Identification and Compensation of Wiener-Hammerstein Systems with Feedback,” *Icassp 2011*, pp. 4056–4059, 2011.
- [8] C. Paleologu, S. Ciochină, J. Benesty, and S. L. Grant, „An overview on optimized NLMS algorithms for acoustic echo cancellation,” *EURASIP Journal on Advances in Signal Processing*, vol. 2015, no. 1, p. 97, 2015. [Online]. Available: <https://asp-urasipjournals.springeropen.com/articles/10.1186/s13634-015-0283-1>
- [9] E. J. Dempsey and D. T. Westwick, „Identification of Hammerstein Models with Cubic Spline Nonlinearities,” *IEEE Transactions on Biomedical Engineering*, vol. 51, no. 2, pp. 237–245, 2004.
- [10] I. M. Iqbal and N. Aziz, „Comparison of various Wiener model identification approach in modelling nonlinear process,” in *Proc. 2011 Data Mining and Optimization Conf.*, no. June,

- 2011, pp. 134–140. [Online]. Available: <http://www.scopus.com/inward/record.url?eid=2-s2.0-80055036827&partnerID=tZOtx3y1>
- [11] T. Mikami, „Detecting Nonlinearity in Prediction Residuals of Snoring Sounds,” in *ICROS-SICE International Joint Conference*, 2009, pp. 5256–5259.
- [12] Z. A. Khan, U. Zabit, O. D. Bernal, M. O. Ullah, T. Bosch, and S. Member, „Adaptive Cancellation of Parasitic Vibrations Affecting a Self-Mixing Interferometric Laser Sensor,” *IEEE Transactions on Instrumentation and Measurement*, vol. 66, no. 2, pp. 332–339, 2017.
- [13] X. Wu, N. Zheng, X. Yang, J. Shi, and H. Chen, „A spline-based hammerstein predistortion for 3G power amplifiers with hard nonlinearities,” *Proceedings of the 2010 2nd International Conference on Future Computer and Communication, ICFCC 2010*, vol. 3, pp. 741–745, 2010.
- [14] E. W. Widrow, Bernard, *Adaptive Inverse Control A Signal Processing Approach*, 2013, vol. 53, no. 9.
- [15] L. Sujbert, „A Filtered Reference – Filtered Error LMS Algorithm.” [Online]. Available: http://dsp.mit.bme.hu/userfiles/publikaciok/sujbert_wisp99b.pdf
- [16] H. Unbehauen, „Identificatin of Nonlinear Systems,” in *Control Systems, Robotics and Automation - Industrial Applications of Control Systems VI.*, 1967, vol. VI. [Online]. Available: <http://pubs.acs.org/doi/pdf/10.1021/ie50690a012>
- [17] O. Nelles, *Nonlinear System Identification*. Berlin: Springer, 2000.
- [18] J. Sjöberg, Q. Zhang, L. Lennart, A. Benveniste, B. Deylon, P.-Y. Glorennec, H. Hjalmarsson, and A. Juditsky, „Nonlinear Black-Box Modeling in System Identification: a Unified Overview,” *Automatica*, vol. 31, no. 12, pp. 1691–1724, 1995.
- [19] L. Ljung, „Approaches to identification of nonlinear systems,” *Control Conference (CCC), 2010 29th Chinese*, pp. 1–5, 2010. [Online]. Available: http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=5572936
- [20] S. A. Billings, *Nonlinear System Identification*. Wiley, 2013, vol. 1, no. 1.
- [21] N. A. Rahim, M. N. Taib, and M. I. Yusof, „Nonlinear system identification for a Dc motor using NARMAX approach,” *2003 Asian Conference on Sensors, AsiaSENSE 2003*, pp. 305–311, 2003.
- [22] S. A. Billings and H.-L. Wei, „A new class of wavelet networks for nonlinear system identification.” *IEEE transactions on neural networks / a publication of the IEEE Neural Networks Council*, vol. 16, no. 4, pp. 862–874, 2005.

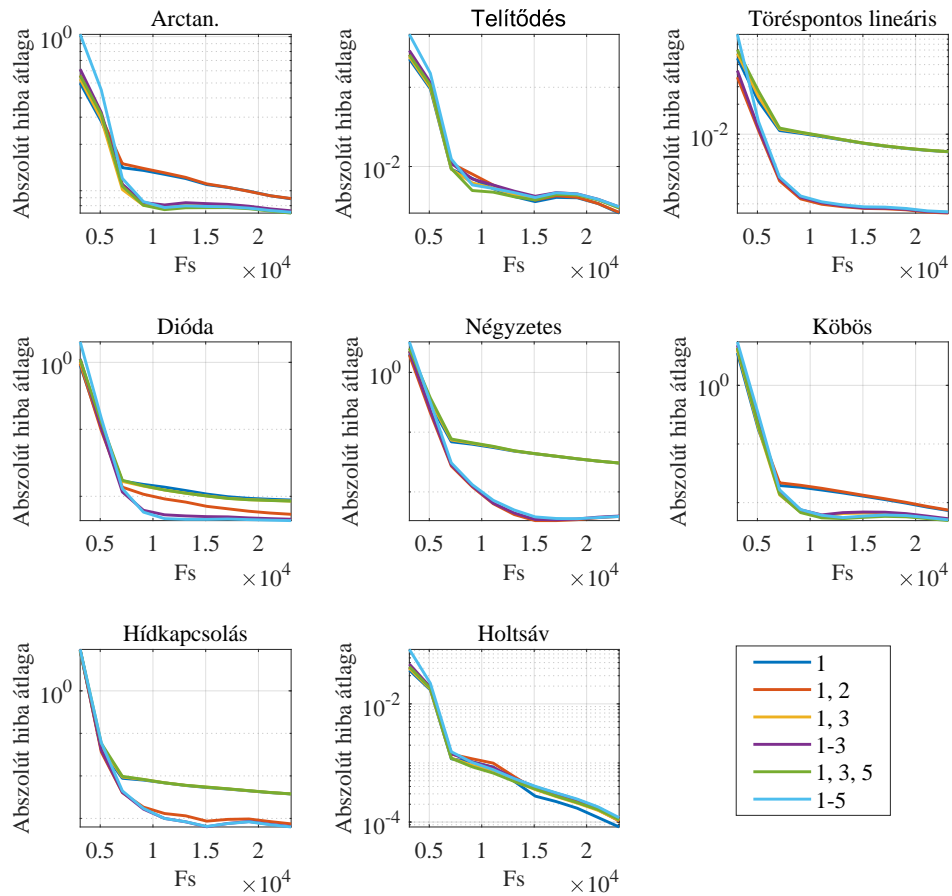
- [23] T. B. Schön, A. Wills, B. Ninness, T. B. Schon, A. Wills, B. Ninness, T. B. Schön, A. Wills, and B. Ninness, „System identification of nonlinear state-space models,” *Automatica*, vol. 37, no. 1, pp. 39–49, 2011. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S0005109810004279>
- [24] P. S. R. Diniz, *Adaptive Filtering - algorithms and practical implementation*, 4th ed. New York: Springer, 2013.
- [25] R. Tanjad and S. Wongsas, „Model structure selection strategy for Wiener model identification with piecewise linearisation,” in *Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology (ECTI) Association of Thailand - Conference 2011*, 2011, pp. 553–556.
- [26] R. Isermann and M. Munchhof, *Identification of Dynamic Systems*, 2003.
- [27] MathWorks, „What are Hammerstein-Wiener Models?” 2017. [Online]. Available: <https://www.mathworks.com/help/ident/ug/what-are-hammerstein-wiener-models.html>
- [28] J. T. Hsu and K. D. T. Ngo, „A Hammerstein-based dynamic model for hysteresis phenomenon,” *IEEE Transactions on Power Electronics*, vol. 12, no. 3, pp. 406–413, 1997.
- [29] A. Hagenblad and A. Hagenblad, „Identification of Wiener Models,” no. July, pp. 1–13, 1998.
- [30] F. Ding, X. P. Liu, and G. Liu, „Identification methods for Hammerstein nonlinear systems,” *Digital Signal Processing: A Review Journal*, vol. 21, no. 2, pp. 215–238, 2011.
- [31] MathWorks, „Available Nonlinearity Estimators for Hammerstein-Wiener Models,” 2017. [Online]. Available: <https://www.mathworks.com/help/ident/ug/nonlinearity-estimators-for-hammerstein-wiener-models.html>
- [32] E. Luiz and O. Batista, „A New Perspective on the Convergence and Stability of NLMS Hammerstein Filters,” in *International Symposium on Image and Signal Processing and Analysis*, no. Ispa, 2013, pp. 336–341.
- [33] J. Voros, „Recursive identification of Hammarstein sytems with polynomial nonlinearities,” *Journal of Electrical Engineering*, vol. 57, no. 1, pp. 42–46, 2006.
- [34] K. Sunat and C. Lursinsap, „Highly successful learning based on modified Catmull-Rom spline activation function,” in *International Joint Conference on Neural Networks*, 2001.
- [35] S. Scardapane, M. Scarpiniti, D. Comminiello, and A. Uncini, „Learning activation functions from data using cubic spline interpolation,” *Nips*, pp. 1–12, 2016. [Online]. Available: <http://arxiv.org/abs/1605.05509>

- [36] L. Vecci, F. Piazza, and A. Uncini, „Learning and approximation capabilities of adaptive spline activation function neural networks,” *Neural Networks*, vol. 11, no. 2, pp. 259–270, 1998.
- [37] C. Yuksel, S. Schaefer, and J. Keyser, „Parameterization and applications of CatmullRom curves,” *CAD Computer Aided Design*, vol. 43, no. 7, pp. 747–755, 2011.
- [38] —, „On the parameterization of Catmull-Rom curves,” *2009 SIAM/ACM Joint Conference on Geometric and Physical Modeling on - SPM '09*, no. c, p. 47, 2009. [Online]. Available: <http://portal.acm.org/citation.cfm?doid=1629255.1629262>
- [39] K. Emancipator and M. H. Kroll, „A quantitative measure of nonlinearity,” *Clinical Chemistry*, vol. 39, no. 5, pp. 766–772, 1993.
- [40] P. Enqvist and J. Karlsson, „Minimal Itakura-Saito distance and covariance interpolation,” *Proceedings of the IEEE Conference on Decision and Control*, no. October, pp. 137–142, 2008.
- [41] IEEE Power & Energy Society, *IEEE Standard Definitions for the Measurement of Electric Power Quantities Under Sinusoidal, Nonsinusoidal, Balanced, or Unbalanced Conditions*, iee std 1 ed., 2010.
- [42] A. Radouane, F. Giri, F. Ikhouane, and F. Z. Chaoui, „Wiener system identification in presence of hysteresis nonlinearity,” in *IFAC Proceedings Volumes (IFAC-PapersOnline)*, vol. 19, 2014, pp. 3214–3219.
- [43] R. Banning and W. L. D. Koning, „Discrete-Time Inverse Hysteresis Control,” in *European Control Conference*, 2001, pp. 342–347.
- [44] R. Dong, Y. Tan, K. Janschek, and X. Chen, „Non-smooth predictive control for mechanical transmission systems with backlash-like hysteresis,” *Nonlinear Dynamics*, vol. 85, no. 4, pp. 2277–2295, 2016.
- [45] F. Giri, Y. Rochdi, and F. Chaoui, „System identification of Hammerstein models including a hysteresis input nonlinearity,” in *IFAC Proceedings Volumes*, vol. 37, no. 12. Elsevier, 2004, pp. 197–202. [Online]. Available: <http://linkinghub.elsevier.com/retrieve/pii/S1474667017314672>
- [46] L. Vanbeylen, R. Pintelon, and P. De Groen, „Blind maximum likelihood Identification of Wiener systems with measurement noise,” in *European Control Conference*, vol. 15, no. PART 1, 2009, pp. 1686–1691.
- [47] J. A. Apolinário and S. L. Netto, „QRD-RLS adaptive filtering,” in *QRD-RLS Adaptive Filtering*, 2009, pp. 23–49.

- [48] M. D. Sawale and R. N. Yadav, „Modified normalized least mean square algorithm with improved minimization criterion,” in *International Conference on Computational Intelligence and Communication Systems*, no. 3, 2011, pp. 541–544.
- [49] A. Hagenblad and L. Ljung, „Maximum Likelihood Identification of Wiener Models with a Linear Regression Initialization,” in *IEEE Conference on Decision & Control*, no. December, 1998, pp. 2–3.
- [50] —, „Maximum likelihood estimation of Wiener models,” in *IEEE Conference on Decision and Control*, vol. 3, 1986, pp. 2417–2418. [Online]. Available: <http://ieeexplore.ieee.org/document/914162/>
- [51] A. Haryanto, A. Turnip, and K. S. Hong, „Parameter Identification of a Superheater Boiler System Based on Wiener-Hammerstein Model using Maximum Likelihood Method,” in *Asian Control Conference*, 2009, pp. 1346–1351.

A. melléklet

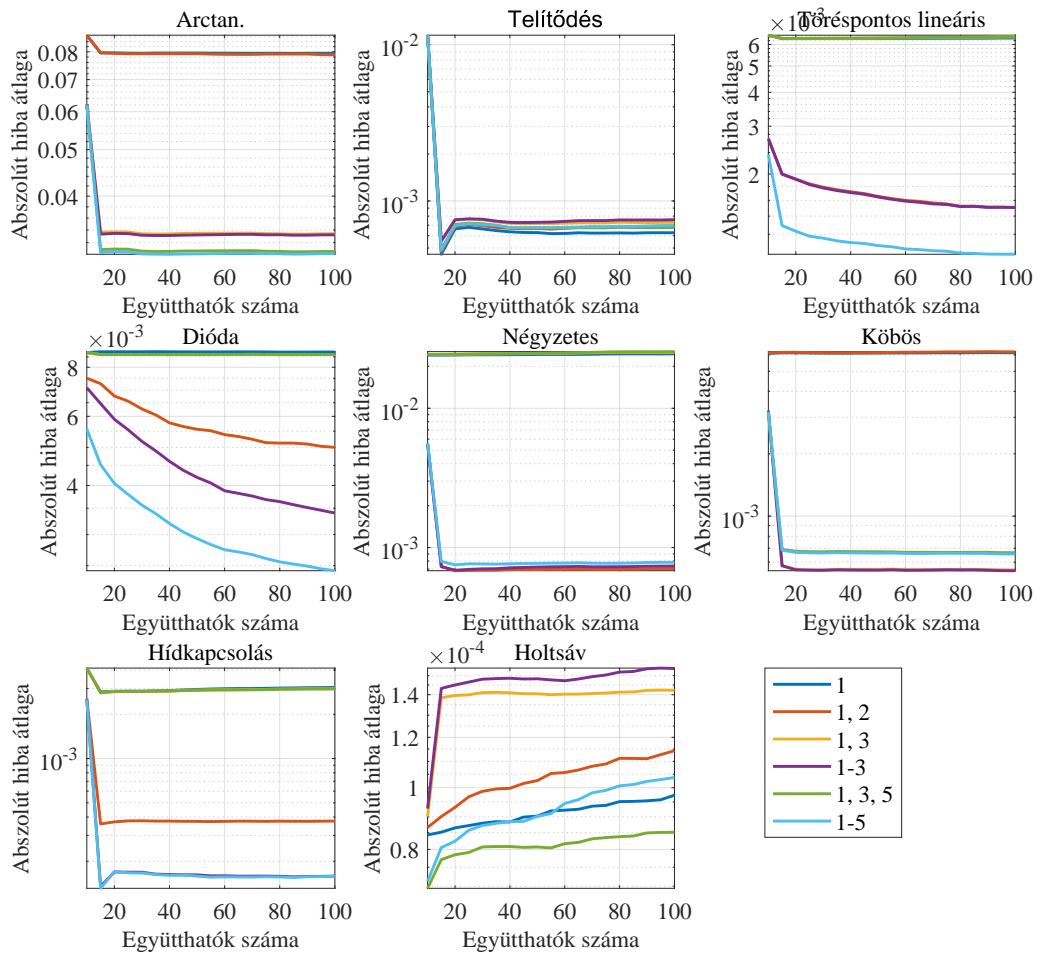
LMS algoritmus átlagos abszolút hibajelének alakulása a 7000. és a 10000. lépés között.



54. ábra. Volterra-modell hibajelének alakulása LMS algoritmus és különféle mintavételi frekvenciák esetén.

B. melléklet

Az RLS algoritmus átlagos abszolút hibajelének alakulása.



55. ábra. Együtthatók számának hatása a Volterra-modell hibájára, RLS algoritmus esetén