



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

Koblász Kolos

**TELJESÍTMÉNYERŐSÍTŐK
KARAKTERISZTIKÁJÁNAK
LINEARIZÁLÁSI MÓDSZEREI**

KONZULENS

Dr. Sujbert László

BUDAPEST, 2017

Tartalomjegyzék

Összefoglaló	5
Abstract.....	6
1. Bevezetés	7
2. Erősítők tulajdonságai és a linearitásuk kvantitatív mértéke	8
2.1 Munkapont szerinti osztályozás	9
2.2 Erősítés.....	11
2.3 Hatásfok	12
2.4 Intermoduláció	12
2.5 AM-AM, AM-PM görbék.....	14
2.6 EVM.....	16
2.7 BER.....	17
2.8 ACPR	17
3. Teljesítményerősítők modellezése	19
3.1 Memóriamentes modellek.....	19
3.1.1 Saleh.....	19
3.1.2 Rapp.....	20
3.2 Memóriával rendelkező modellek	20
3.2.1 Volterra sorok	20
3.2.2 Hammerstein	21
3.2.3 Wiener.....	21
3.2.4 Memória Polinom	22
4. Linearizálási technikák	23
4.1 Analógmódszerek	23
4.1.1 Túlméretezés.....	23
4.1.2 Dinamikus munkapont beállítás.....	24
4.1.3 Burkoló leválasztása és visszaállítása.....	24
4.1.4 Cartesian visszacsatolás	25
4.1.5 Előreccatolásos linearizálás	26
4.1.6 Analóg előtorzítás	29
4.2 Digitális előtorzítók	30
4.2.1 A jelfeldolgozó láncban betöltött hely szerinti felosztás.....	32

4.2.2 Memória szerinti felosztás	32
4.2.3 Adaptáció szerinti felosztás	33
4.2.4 Az előtorzítás kivitelezése szerinti felosztás	33
5. Matematikai összefoglaló	35
5.1 Legkisebb négyzetek módszere	35
5.2 Polinomiális regresszió	36
5.3 Pseudóinverz	37
5.4 NLMS algoritmus	37
5.5 Kereszt korrelációs időkülönbség korrekció.....	38
6. Előtorzítók paraméter becslése.....	40
6.1 Inverz tanuló architektúra	40
6.2 Direkt tanuló architektúra	42
7. Szimulációk.....	44
7.1 Szimulációk Inverz Tanuló Architektúrával	45
7.2 Szimulációk Direkt Tanuló Architektúrával	50
8. Implementáció.....	53
8.1 Az AD9361-es RF Chip tulajdonságai	54
8.2 A Zynq Z-7020-as SoC tulajdonságai és a fejlesztés menete	59
8.3 Az FMCOMMS-3 panel illesztése ZedBoardhoz.....	61
8.4 Egyedi perifériák készítése	64
8.5 A beágyazott szoftver	69
9. Mérések, előtorzítás	72
10. Értékelés, tovább fejlesztési lehetőségek.....	76
Irodalomjegyzék.....	77

HALLGATÓI NYILATKOZAT

Alulírott Koblász Kolos, szigorló hallgató kijelentem, hogy ezt a diplomatervet meg nem engedett segítség nélkül, saját magam készítettem, csak a megadott forrásokat (szakirodalom, eszközök stb.) használtam fel. Minden olyan részt, melyet szó szerint, vagy azonos értelemben, de átfogalmazva más forrásból átvettem, egyértelműen, a forrás megadásával megjelöltem.

Hozzájárulok, hogy a jelen munkám alapadatait (szerző(k), cím, angol és magyar nyelvű tartalmi kivonat, készítés éve, konzulens(ek) neve) a BME VIK nyilvánosan hozzáférhető elektronikus formában, a munka teljes szövegét pedig az egyetem belső hálózatán keresztül (vagy hitelesített felhasználók számára) közzétegye. Kijelentem, hogy a benyújtott munka és annak elektronikus verziója megegyezik. Dékáni engedéllyel titkosított diplomatervek esetén a dolgozat szövege csak 3 év eltelte után válik hozzáférhetővé.

Kelt: Budapest, 2017. 05. 10.

.....
Koblász Kolos

Összefoglaló

Ez a diplomaterv nagyfrekvenciás erősítők linearizálásával, azon belül is a digitális előtorzítás témakörével foglalkozik.

Diplomatervem első fejezetében vázolom a modern rádiófrekvenciás és mikrohullámú távközlő hálózatok erősítőivel kapcsolatos kihívásokat, elvárásokat. Bemutatom, miért okoz gondot a nem konstans burkolójú modulációkhoz erősítőt tervezni, ha annak hatásfoka is jó kell, hogy legyen.

A második fejezetben röviden bemutatom a rádiófrekvenciás és mikrohullámú erősítők főbb tulajdonságait és e tulajdonságok kvantitatív jellemzésére szolgáló mérőszámokat.

A harmadik fejezetben számos erősítő linearizálási stratégiát mutatok be, kiemelt figyelmet fordítva a digitális előtorzításra. Kitérek ezek felhasználási területére, képességeikre (hatásfok, linearizálás mértéke, sáv szélesség, bonyolultság).

A negyedik fejezetben nemlineáris, főképp polinom alapú, erősítő modellek ismertetésével foglalkozok.

Az ötödik fejezetben a memória polinomon alapuló erősítő modell inverzének az azonosításához szükséges matematikai ismereteket foglalom röviden össze.

A hatodik fejezetben az Inverz Tanuló Architektúra és a Direkt Tanuló Architektúra szerinti előtorzító azonosítását írom le.

A hetedik fejezetben az eddigi ismeretek alapján készült Matlab szimulációk eredményeit írom le.

A nyolcadik fejezetben részletesen bemutatom az általam tervezett és implementált Zynq SoC alapú előtorzítót.

A kilencedik fejezetben a mérések menetét és eredményeit ismertetem.

A tizedik fejezetben értékelem a projektet és vázolom a továbbfejlesztési lehetőségeket.

Abstract

This thesis is about high frequency amplifier linearization, focusing on digital predistortion.

In the first chapter I show the challenges of high frequency power amplifier design for state of the art telecommunication networks. The explanation of amplifier design problems associated with non constant envelop signals is given.

In the second chapter I briefly present the most important properties of radio frequency and microwave amplifiers, including quantitative figures of merit in terms of linearity.

In the third chapter I present numerous amplifier linearization strategies, concentrating on digital predistortion. Details concerning the field of application, capabilities (efficiency, linearization performance, bandwidth, complicity) are also given.

In the fourth chapter I show the most important nonlinear models used for power amplifier characterization.

In the fifth chapter mathematical tools are presented for the aforementioned characterization process.

The sixth chapter goes into the details of Inverse and Direct Learning Architecture usage regarding predistorters.

The seventh chapter contains the results of my predistorter simulations. These simulations were done in Matlab.

In the eight chapter I give a detailed description of my baseband digital predistorter using a Zynq SoC and an AD9361 transreceiver RF IC.

The ninth chapter contains the measurement results of the predistorter.

In the tenth chapter progress evaluation and discussion of opportunities regarding the project's future is given.

1. Bevezetés

A rádióhullámú kommunikációs hálózatok fontos alkotóelemei a teljesítményerősítők. Ezek az eszközök felelősek a kimenő jelszint kellő értékre növeléséért. Bár funkciójuk igen egyszerű, realizációjuk rendkívül bonyolult, tulajdonságaik nagyban meghatározzák a teljes kommunikációs rendszer tulajdonságait, képességeit. Üzemeltetésük költséges ezért cél, hogy minél jobb hatásfokkal rendelkezzenek, de kritikus szempont az erősítők nemlineáris torzítása is. Azért fontos, hogy az erősítők lineáris karakterisztikájúak legyenek a használt teljesítmény tartományban, mert így elkerülhető a különböző frekvenciájú jelkomponensek intermodulációjából származó bit-hibaarány (BER) illetve a kiszélesedő spektrum okozta szomszédos csatornateljesítmény-arány (ACPR) növekedése. Előbbi a saját kommunikációs csatornánk átviteli kapacitását csökkenti, utóbbi pedig a szomszédos csatornák kommunikációját rontja, lehetetleníti el. A hatásfok, a linearitás, a gyártási és tervezési költségek nem javíthatóak egymástól függetlenül. A legnagyobb problémát a hatásfok és a linearitás kapcsolata okozza. A jó hatásfokú erősítő nem lineáris, a lineáris erősítő rossz hatásfokú. E két tulajdonság egyensúlyban tartására számos analóg és digitális megoldás született a rádiózás hajnala óta. Ilyen megoldásokat fogok a továbbiakban bemutatni, kiemelt hangsúlyt fektetve az alapsávi digitális előtorzítás módszerére.

2. Erősítők tulajdonságai és a linearitásuk kvantitatív mértéke

A nagyfrekvenciás erősítőket számos módon lehet osztályozni. Egy ilyen szempont lehet, hogy milyen elektronikai alkatrész felel a jelszint növelésért. Léteznek félvezető alapú erősítők (Solid State Power Amplifier - SSPA) és elektroncső alapúak (Tube Amplifier -TA). Elektroncső alapú erősítőket ma már csak rendkívül speciális esetekben használnak. Ilyen speciális esetek a haladó hullámú erősítő csővel (Traveling Wave Tube Amplifier - TWTA) készült műholdfedélzeti erősítők és a radartechnikában használt Klystron cső alapú erősítők. A mobil kommunikációs hálózatokban kizárólag SSPA-kat használnak. Dolgozatomban ilyen eszközökkel fogok foglalkozni, bár sok megoldás alkalmazható elektroncsövekre is.

További osztályozási szempont az erősítők jelszintje. Itt megkülönböztetünk kiszajú (lineáris) erősítőket és teljesítmény erősítőket (nemlineáris). Egy beérkező zajos jel erősítése növelni fogja a zajt is és a hasznos jelet is, azonban az erősítő lánc még a saját zajával is növelni fogja a végső zajteljesítményt. A 2.1 egyenlet alapján látható, hogy az erősítő lánc első tagjai kritikusak ebből e szempontból. Mivel itt kis jelszinteken dolgozunk, a különböző félvezetőket nem vezéreljük ki, ezért nem kell a tervezés során linearitásukkal foglalkozni.

$$F_{ered \delta} = F_1 + \sum_{i=2}^N \frac{F_i - 1}{\prod_{k=1}^{i-1} G_k} \quad (2.1)$$

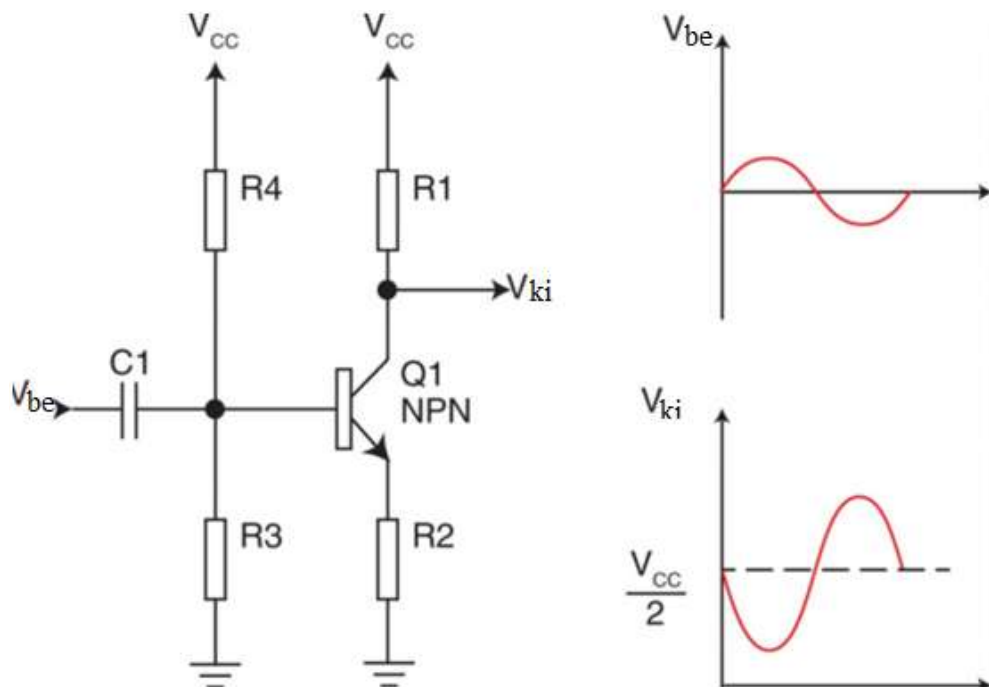
Teljesítményerősítők esetén azonban szükséges azok linearitásával is törődni. Zajuk nincs nagy hatással a teljes lánc zajtényezőjére, hisz a végfok az utolsó erősítő az antenna előtt. Ezeken az eszközökön múlik a késztermék áramfelvétele, hatásfoka, kimenőszintje, linearitása és sokszor az ára is. A teljesítmény erősítőket osztályokba lehet sorolni annak függvényében, hogy milyen DC munkapontban üzemeltetjük őket, és hogy milyen kapcsolást valósítunk meg.

2.1 Munkapont szerinti osztályozás

Forrás:[1]

A osztály

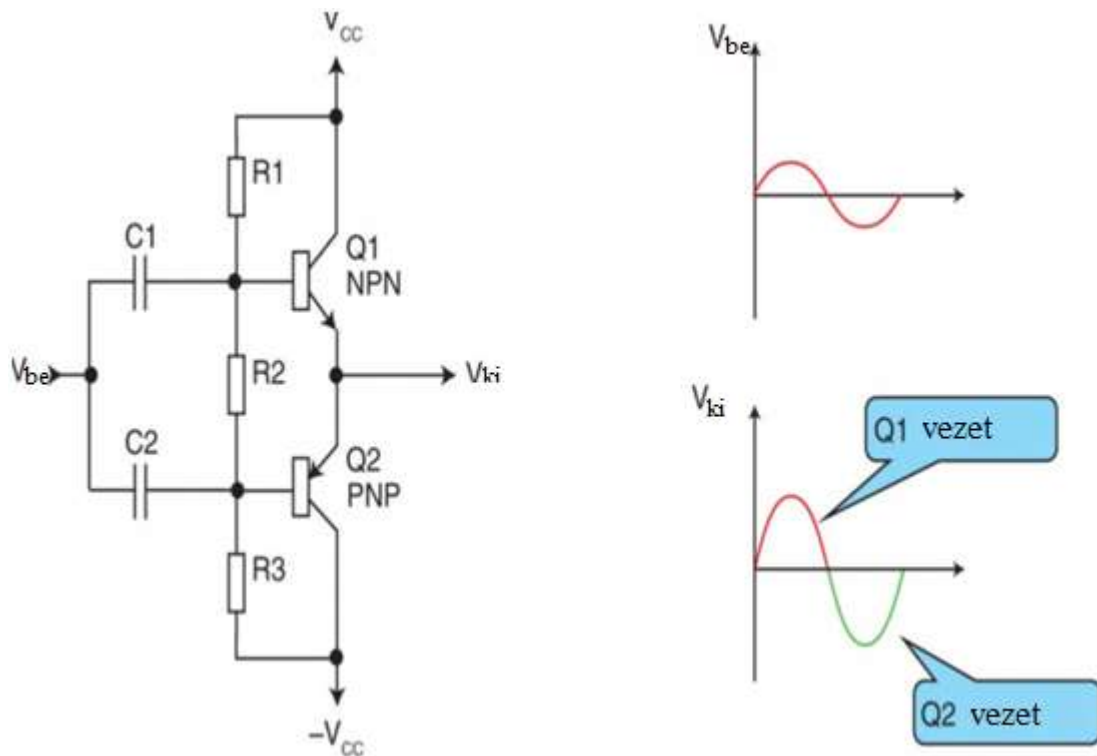
Ez a legegyszerűbb erősítő osztály. A tranzisztoron olyan munkapontot állítunk be, hogy folyamatosan folyjon rajta áram, és kollektor feszültsége közelítően a tápfeszültség fele legyen. Így a tranzisztor vezetési szöge 360° és kevéssé torzít. Sajnos a hatásfoka még elméletileg sem lehet több 50%-nál, a gyakorlatban pedig inkább 30% körül szokott alakulni még szerencsés esetben is. Kapcsolás technikailag rendkívül egyszerű és a többi osztályhoz képest kisebb torzítási jellemzőkkel bír.



2.1. ábra: A osztályú alapkapsolás.

B osztály

A B osztályú erősítők komplementer tranzisztorokat használnak melyek push-pull üzemben működnek. A tranzisztorokat olyan munkapontba kell állítani, hogy az egyik a jel pozitív, a másik a negatív félperiódusában vezessen. Az A osztályhoz képest jobb a hatásfok, de a nulla átmenetknél jelentősen torzul a jel.



2.2. ábra: B osztályú alapkapsolás.

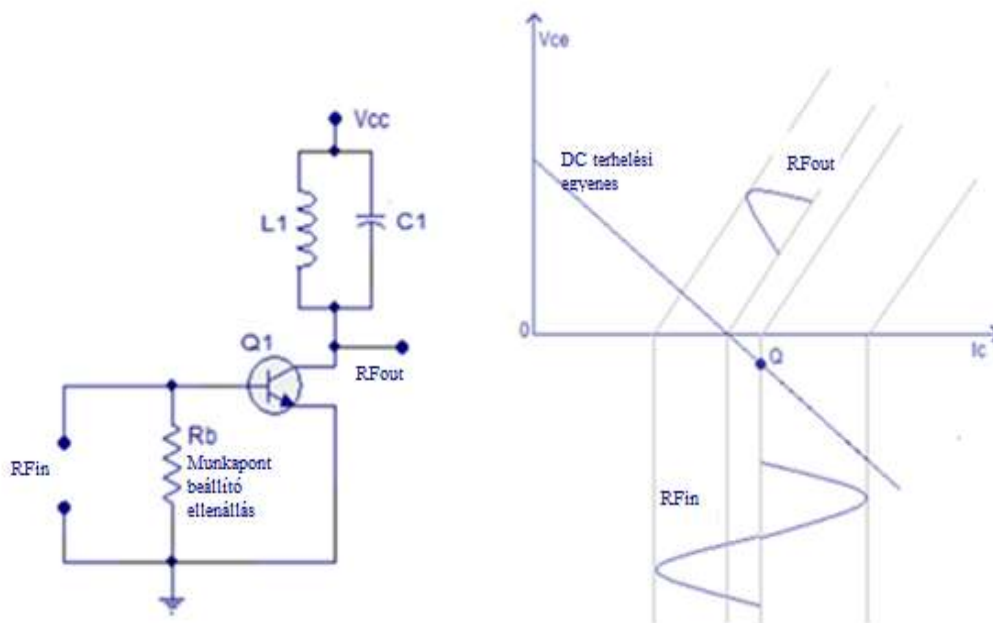
AB osztály

Az AB osztályú erősítők olyan B osztályú erősítők, melyekben a tranzisztorok kicsit eltérő munkapontban üzemelnek így még akkor is folyik rajtuk valamennyi áram, amikor épp nulla a vezérlő jel pillanatnyi értéke. Torzításuk és hatásfokuk is a két osztály között helyezkedik el.

C osztály

Forrás:[2]

Olyan DC munkapontba kell állítani az erősítőket, hogy vezetési szögük kisebb legyen, mint 180° . Kizárólag RF jelek erősítésére használhatóak mert erősen nemlineárisak. Azon modulációk esetén használhatóak, ahol a jel amplitúdója nem hordoz információt, csak fázisa illetve frekvenciája. Hatásfokuk elérheti a 90%-ot.



2.3. ábra: C osztályú alapkapsolás.

Ezek az osztályok kívül vannak további kapcsolások is, de mobil bázisállomásokban jellemzően A vagy AB-t szoktak alkalmazni. Ha végfok erősítőt tervezünk, mindig szem előtt kell tartani, hogy mekkora linearitást és hatásfokot várunk el az eszköztől, és ennek függvényében kiválasztani azt az erősítő osztályt, ami képes lesz teljesíteni a specifikációt.

2.2 Erősítés

Az egyik legfontosabb tulajdonsága az erősítőknek. Frekvencia, teljesítmény és hőmérsékletfüggő, az idő múlásával is változik. Az eszköz azon képességét adja meg, hogy hányszorosára képes növelni a beérkező jel teljesítményét a kimeneten, illetéssel lezárások esetén. Ideális esetben a használni kívánt frekvenciatartományban konstans értékű.

$$G = \frac{P_{out}}{P_{in}} = \frac{V_{in}^2 / Z_0}{V_{out}^2 / Z_0} = \frac{V_{in}^2}{V_{out}^2}, \quad G^{dB} = 10 \log_{10} \left(\frac{P_{out}}{P_{in}} \right) \quad (2.2)$$

2.3 Hatásfok

A hatásfok azt jelenti, hogy a tápegységből felvett teljesítmény mekkora részben konvertálódik át jelteljesítménnyé. Minél közelebb áll a 100%-hoz, annál kisebb része fog hővé alakulni a DC teljesítmények. Mobilhálózatok bázis állomásai esetén ez a hő pénzben mérhető. Az áramfelvétel adja egy bázisállomás működési költségének legnagyobb hányadát. Ennek a villamos áramfogyasztásnak a nagyját a végfokok illetve a hűtésükre használt légkondicionáló rendszer működtetése teszi ki.

$$\eta = \frac{P_{out}}{P_{DC}} \quad (2.3)$$

2.4 Intermoduláció

Forrás:[3]

Az intermoduláció azt jelenti, hogy a jelkomponensek összeszorzódnak egymással szándékunkon kívül.

Tekintsük a következő nemlineáris rendszert és az $x(t)$ gerjesztést:

$$y(t) = a_1x(t) + a_2x(t)^2 + a_3x(t)^3 \quad (2.4)$$

$$x(t) = A \cos(\omega t). \quad (2.5)$$

A rendszer válasza ebben az esetben:

$$y(t) = a_1A \cos(\omega t) + a_2(A \cos(\omega t))^2 + a_3(A \cos(\omega t))^3 \quad (2.6)$$

$$y(t) = a_1A \cos(\omega t) + \frac{a_2A^2}{2}(1 + \cos(2\omega t)) + \frac{a_3A^3}{4}(3 \cos(\omega t) + \cos(3\omega t)) \quad (2.7)$$

$$y(t) = \frac{a_2A^2}{2} + \left(a_1A + \frac{3a_3A^3}{4}\right) \cos(\omega t) + \frac{a_2A^2}{2} \cos(2\omega t) + \frac{a_3A^3}{4} \cos(3\omega t) \quad (2.8)$$

Ha egy nemlineáris rendszerre szinuszos gerjesztést adunk, akkor megjelennek felharmonikus jelkomponensek is, melyek amplitúdója a rendszer paramétereitől függ. Ezek a paraméterek hőmérséklet- és időfüggőek. Nézzük meg, hogy mit okoz, ha kettő szinuszból álló jelet kapcsolunk az előbbi rendszerre:

Gerjesztés:

$$x(t) = A_1 \cos(\omega_1 t) + A_2 \cos(\omega_2 t) \quad (2.9)$$

Válasz:

DC komponensek:

$$y(t) = \frac{1}{2} a_2 (A_1^2 + A_2^2) + \quad (2.10)$$

Elsőfokú tagok

$$\begin{aligned} & \left[a_1 A_1 + \frac{3}{4} a_3 A_1 (A_1^2 + 2A_2^2) \right] \cos(\omega_1 t) + \\ & \left[a_1 A_2 + \frac{3}{4} a_3 A_2 (2A_1^2 + A_2^2) \right] \cos(\omega_2 t) + \end{aligned} \quad (2.11)$$

Másodfokú tagok

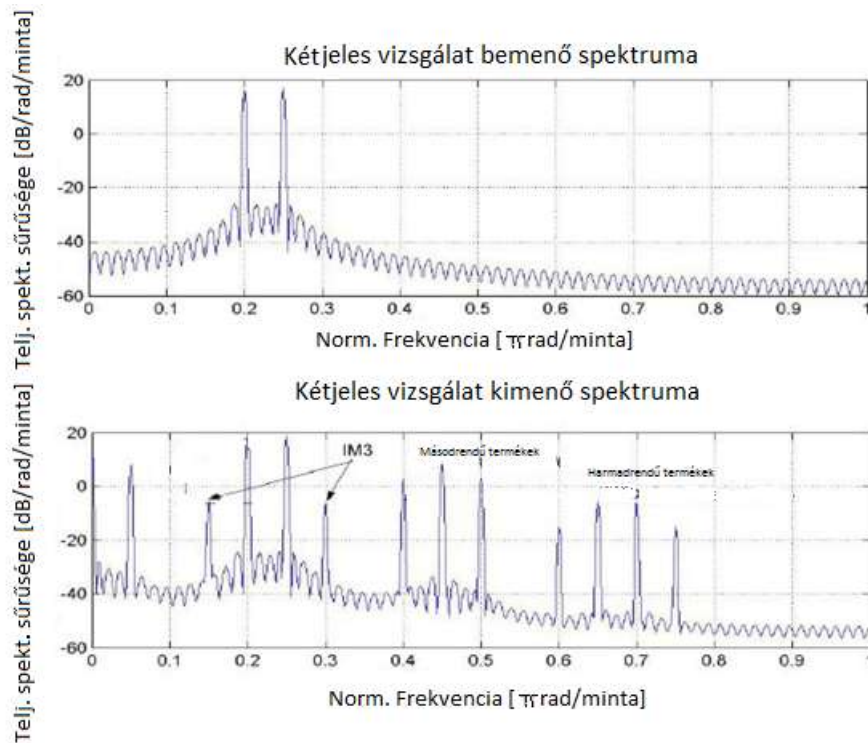
$$\begin{aligned} & 0.5 a_2 [A_1^2 \cos(2\omega_1 t) + A_2^2 \cos(2\omega_2 t)] + \\ & a_2 A_1 A_2 [\cos((\omega_1 + \omega_2)t) + \cos((\omega_1 - \omega_2)t)] + \end{aligned} \quad (2.12)$$

Harmadfokú tagok

$$\begin{aligned} & 0.25 a_3 [A_1^3 \cos(3\omega_1 t) + A_2^3 \cos(3\omega_2 t)] + \\ & 0.75 a_3 A_1^2 A_2 [\cos(2\omega_1 + \omega_2)t + \cos((2\omega_1 - \omega_2)t)] + \\ & 0.75 a_3 A_2^2 A_1 [\cos(2\omega_2 + \omega_1)t + \cos((2\omega_2 - \omega_1)t)] \end{aligned} \quad (2.13)$$

Az RF mérés technikában ezt a gerjesztést kétjeles vizsgálatnak hívják és minden telekommunikációs szabvány előírásokat fogalmaz meg a harmadrendű tagok abszolút és gerjesztő jelhez képesti relatív teljesítményére. A nulladfokú tagokat egy DC blokkoló kondenzátorral könnyedén ki lehet szűrni bármely erősítő kimenetéről. Az elsőfokú tagok a tulajdonképpeni erősítés. A másodfokú tagok frekvenciái sávon kívülre esnek, tehát olyan nagy frekvenciájúak, hogy szűrésük kivitelezhető vagy pusztán az áramkör képtelen átvinni őket. A problémát a harmadfokú tagok okozzák. Ezeket hívjuk harmadrendű intermodulációs termékeknek, túl közel esnek a használt frekvenciasávhoz ahhoz, hogy analóg szűrőkkel ki lehessen szűrni őket.

Forrás:[4]



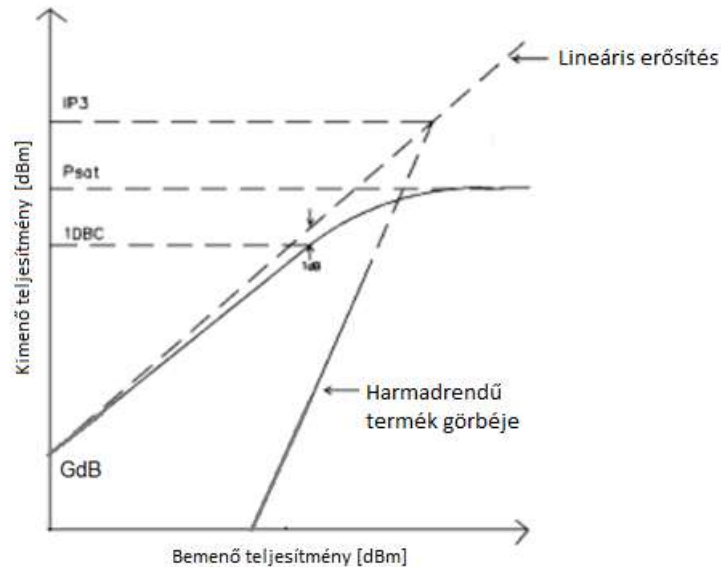
2.4. ábra: Kétjeles vizsgálat teljes spektrumképe.

A 2.4-es ábrán egy általános nemlineáris rendszer, kétjeles gerjesztésre adott válaszát látjuk. Ez a jelenség egy digitálisan modulált jel esetén a spektrum kiszélesedését fogja okozni.

2.5 AM-AM, AM-PM görbék

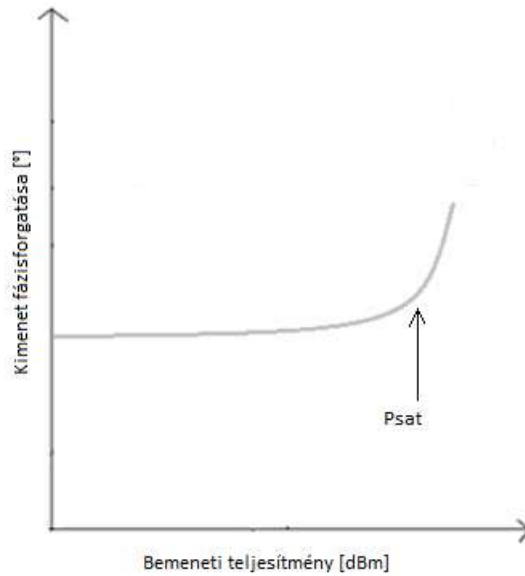
Az AM-AM (Amplitude Modulation to Amplitude Modulation) görbék a nemlineáris erősítők azon tulajdonságát szemléltetik, hogy adott bemeneti jelteljesítményhez mekkora kimeneti jelteljesítmény tartozik. Mind két tengelyt dB-ben szokás skálázni így ideális erősítő esetén egy 45° -os egyenest kapnánk, ami az y-tengelyt G^{dB} –nél metszi. A következő ábrán látható, hogy a bejövő jel növelésével a kimeneti jelszint nem nő arányosan így az eszköz nemlineárisan viselkedik. Tehát az erősítés egy, az adott félvezetőre jellemző, jelszint fölött el kezd csökkenni. Tranzisztorok linearitásának objektív összehasonlítására használunk három nevezetes teljesítmény értéket. A P_{out}^{1dB} pont jelzi azt a jelteljesítményt, aminél már egy dB-lel kisebb a tényleges kimenőszint, mint ideális esetben lenne. Az IP3 pont jelzi azt a kimenő vagy bemenő jelszintet (egyértelműen, de vegyesen használják az adatlapok)

aminél a lineáris erősítéshez tartozó egyenes metszi a harmadrendű intermodulációs termékekhez tartozó egyenest. Ez a pont a valóságban nem kimérhető rendszerint, mivel ilyen vezérlésnél a félvezető már tönkremenne. Minél nagyobb ez az érték annál lineárisabb a tranzisztor. Az utolsó érték a szaturációs szint. Ez az a teljesítmény, aminél többet nem lehet kinyerni a félvezetőből.



2.5. ábra: AM-AM görbe.

Az AM-PM (Amplitude Modulation to Phase Modulation) görbe azt mutatja be, hogy a jelszint növekedésével változik az erősítő fázisforgatása.



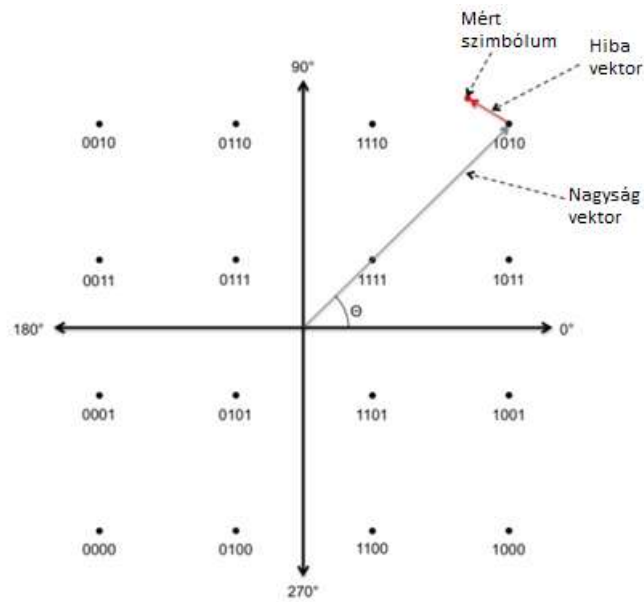
2.6. ábra: AM-PM görbe.

Tehát az erősítők nemlinearitása elrontja mind a kimenő jelszintet mind a fázist ráadásul új nemkívánatos jelkomponensek is bekerülnek a jelbe.

2.6 EVM

A jel amplitúdójának és fázisának torzulása szemléltethető egy digitális moduláció konstellációs diagramján. Ha a végfok nem tudja lineárisan erősíteni minden konstellációs ponthoz tartozó szimbólumot, akkor a mért pontok arrébb fognak esni az ideális helyüktől, így a hiba vektor nagysága megnő. Ez csökkenti az átvitel bitsebességét, mivel a vevő így többet hibázik, ezáltal szükségessé teszi csomagok újraküldését.

Forrás:[5]



2.7. ábra: EVM - Hiba Vektor Nagysága.

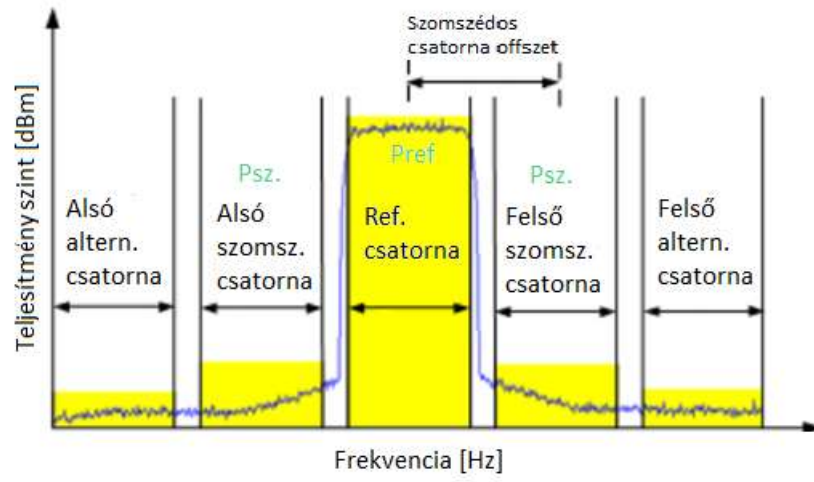
2.7 BER

A bithibaarány (Bit Error Rate) a teljes kommunikációs lánc minőségét jellemzi. Minél kisebb annál több ép adatot sikerül továbbítanunk. Ha túl nagy az EVM, akkor többet hibázik a vevő oldali döntő áramkör, és megnövekszik a BER. Két okból törekszünk nem konstans burkolójú modulátorok erősítőinek linearitását megtartani: az egyik a kommunikációnk BER-jének alacsonyan tartása, a másik a szomszédos csatornák kommunikációjának biztosítása.

2.8 ACPR

Az ACPR (Adjacent Channel Power Ratio) azt fejezi ki, hogy egy kommunikációs csatorna használata során mekkora jelteljesítményt sugárzunk a szomszédos kommunikációs csatornába. Ez a jelteljesítmény a szomszédos átviteli csatorna zajához adódik hozzá így csökkentve a jelzaj viszonyt.

Forrás: [6]



2.8. ábra: Szomszédos csatornák teljesítmény aránya.

3. Teljesítményerősítők modellezése

A 2.4. fejezetben bemutatásra került, hogy egy nemlineáris rendszer milyen módon változtatja meg a jel spektrumát. A jelfeldolgozás területén dolgozó mérnöktudósok régóta próbálnak minél pontosabb, de kezelhető nemlineáris modelleket alkotni. Ezek közül az erősítők modellezésére használtak egy részhalmazát fogom ismertetni. Megkülönböztetünk memóriamentes és memóriával rendelkező modelleket. Ha az erősítőnk keskenysávú, akkor egy memóriamentes modell is kielégítő pontosságú lehet. Ha a rádió spektrum egy széles tartományában kívánjuk üzemeltetni az erősítőt, akkor viszont szükségessé válik figyelembe venni az impedancia illesztő, munkapont beállító dinamikus elemek illetve a félvezető belső kapacitásainak hatását is.

3.1 Memóriamentes modellek

Forrás:[4]

3.1.1 Saleh

A Saleh modell β_a értékű vezérlés hatására szaturálja a kimenetet. α_a a kisjelű erősítés. A maximális fázistolás $\frac{\alpha_p}{\beta_p}$ értékű.

Erősítés:

$$G(|x|) = \begin{cases} \frac{\alpha_a |x|}{1 + \beta_a |x|^2} & , ha |x| \leq \frac{1}{\sqrt{\beta_a}} \\ \frac{\alpha_a}{2\sqrt{\beta_a}} & , ha |x| > \frac{1}{\sqrt{\beta_a}} \end{cases} \quad (3.1)$$

Fázistolás:

$$\psi(|x|) = \frac{\alpha_p |x|^2}{1 + \beta_p |x|^2} \quad (3.2)$$

3.1.2 Rapp

Ez a modell csak az amplitúdó átvitelt modellezi, a fázistolást nem. Három paramétere van: α a kisjelű erősítés, β szaturációs együttható és s a simasági tényező. Ez irányítja a görbe simaságának átmenetét a lineáris és a szaturációs tartomány között.

$$G(|x|) = \frac{\alpha|x|}{\left[1 + \left(\frac{\alpha|x|}{\beta}\right)^{2s}\right]^{\frac{1}{2s}}} \quad (3.3)$$

3.2 Memóriával rendelkező modellek

3.2.1 Volterra sorok

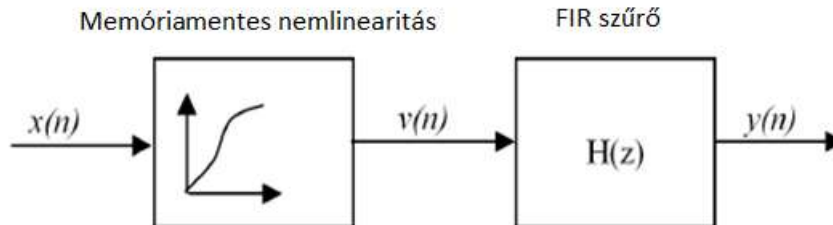
Forrás:[7]

A nemlineáris rendszerek modellezésére használt legáltalánosabb és legszámítás-igényesebb modell a Volterra sor. A rendszer válaszát (ha memóriáját K -val, nemlinearitását L -el, a k_l méretű együttható mátrixokat pedig a_l -val jelöljük) az alábbi egyenlet adja:

$$\begin{aligned} y(n) &= a_0 \\ &+ \sum_{k_1=0}^K a_1(k_1)x(n-k_1) \\ &+ \sum_{k_1=0}^K \sum_{k_2=0}^K a_2(k_1, k_2)x(n-k_1)x(n-k_2) \\ &+ \dots \\ &+ \sum_{k_1=0}^K \sum_{k_2=0}^K \dots \sum_{k_L=0}^K a_L(k_1, k_2, \dots, k_L)x(n-k_1)x(n-k_2) \dots x(n-k_L). \end{aligned} \quad (3.4)$$

3.2.2 Hammerstein

A Hammerstein modell egy kétrészes modell, ami statikus nemlineáris részből és az azt követő FIR szűrőből áll. A modell $L+1$ rendű nemlinearitást és K mélységű memóriát tud kezelni. A Volterra modell egyszerűsítése.



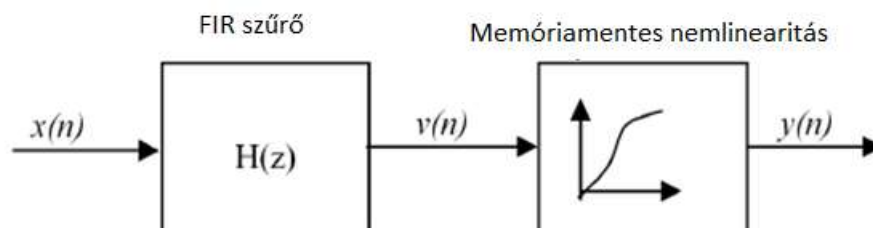
3.1. ábra: Hammerstein modell.

$$v(n) = \sum_{l=0}^L b_l x(n) |x(n)|^l \quad (3.5)$$

$$y(n) = \sum_{k=0}^K c_k v(n-k) \quad (3.6)$$

3.2.3 Wiener

Fordított elrendezésben tartalmazza a Hammerstein modell alkotóelemeit. Identifikálása bonyolultabb a Hammerstein modellnél.



3.2. ábra: Wiener modell.

$$v(n) = \sum_{k=0}^K c_k x(n-k) \quad (3.7)$$

$$y(n) = \sum_{l=0}^L b_l v(n) |v(n)|^l \quad (3.8)$$

3.2.4 Memória Polinom

Ez is a Volterra modell egyszerűsített változata. Széles körben használt teljesítményerősítők modellezésére. Jó kompromisszum a modell pontossága és komplexitása közt. A modell $L+1$ rendű nemlinearitást és K mélységű memóriát tud kezelni.

$$y(n) = \sum_{k=0}^K \sum_{l=0}^L a_{kl} x(n-k) |x(n-k)|^l \quad (3.9)$$

4. Linearizálási technikák

Mivel nem konstans burkolójú modulációval adott sáv szélesség mellett nagyobb adatátviteli sebességet lehet elérni, ezért a modern távközlési szabványok (UMTS, LTE) a jel amplitúdóját is változtatják. Egy, a modulációra jellemző, mérőszám a csúcstényező. Az OFDM (Orthogonal Frequency Division Multiplexing) alapú kommunikáció (LTE) esetén a csúcstényező (pillanatnyi csúcsteljesítmény és átlagteljesítmény hányadosa) igen nagy lehet, mely olyan erősítőt kíván, ami ezeket a csúcsokat is képes torzítás nélkül átvinni. Tehát lesz egy jelünk, amit lineárisan kell erősítenünk, mert nem csak fázisa/frekvenciája, hanem amplitúdója is hordoz információt és átlag teljesítményénél három, hat, de akár ennél is több decibellel nagyobb csúcsokat kell kezelnie a végfoknak. A követelmények első fele nem teszi lehetővé C osztályú erősítő használatát, a második fele pedig az így elérhető határfokot is tovább csökkenti, mivel erősen túl kell méretezni a végfokot a csúcsok lineáris átviteléhez. A határfok és linearitás egyensúlyban tartásához valamilyen módon linearizálni kell a bázisállomások nagy áramfelvételű erősítőit.

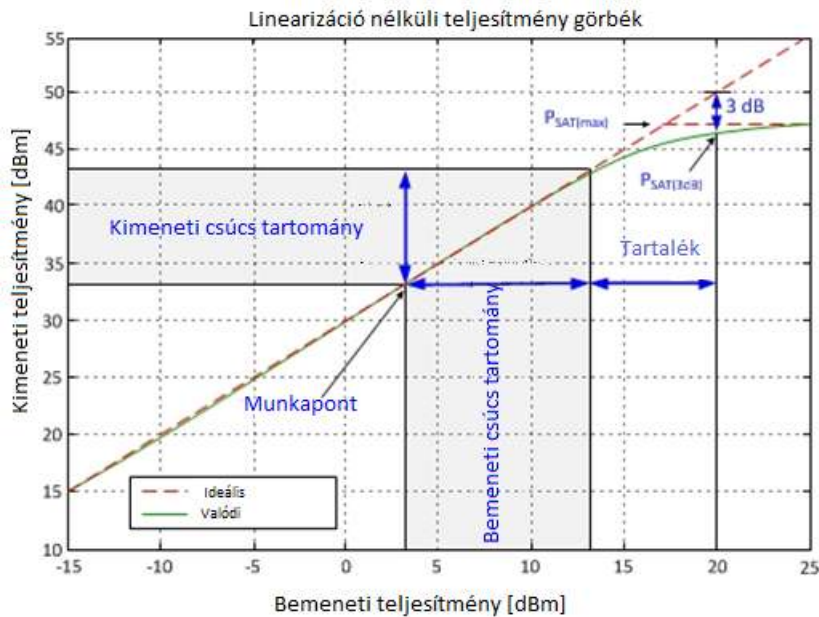
4.1 Analóg módszerek

Azokat a linearizálási technikákat tekintjük analóg linearizálási módszereknek, amik nem mintavételezik a jelet. Ettől függetlenül sokszor szükség lehet mikrokontrollerekre alkalmazásukhoz. Hangsúlyozom, hogy a digitális áramkörök ilyenkor vezérlési és nem jelfeldolgozó szerepet töltenek be.

4.1.1 Túlméretezés

Ez a probléma legegyszerűbb megoldása. Akkor a teljesítmény tartalékkal (Back-Off) tervezzük meg az RF végfokot, hogy a rövid idejű, nagyteljesítményű csúcsok sem viszik szaturációba azt. Előnye a rövidebb tervezési idő, de határfoka rendkívül alacsony. További hátránya, hogy drága. Két dolog tudja jelentősen megdrágítani egy félvezető árát: a nagy frekvencia és a nagy teljesítmény. Távközlési elektronika esetén az előbbi adott a másodikat mi növeljük a szükséges többszörösére. Sokszor a túlméretezés ellenére sem tudja az eszköz teljesíteni az elvárt linearitást.

Forrás:[8]



4.1. ábra: Túlméretezett erősítő karakterisztikája.

4.1.2 Dinamikus munkapont beállítás

Forrás:[9]

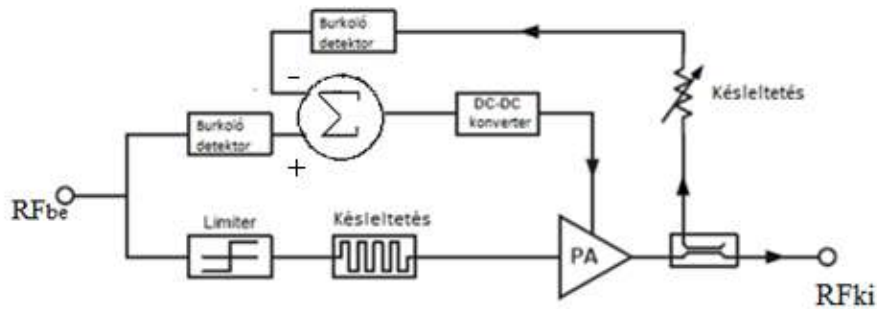
A módszer azon az elven alapszik, hogy az RF jel burkolójának függvényében dinamikusan változik a FET gate elektródájának munkaponti feszültsége. Ha nagy, a burkoló pillanatnyi értéke nagyobb DC feszültséget kap a FET, ha kisebb, akkor a gate munkaponti feszültsége is csökken. Ezzel a módszerrel lehetséges javítani az erősítő hatásfokát, mert az áramfelvétele csak akkor nagy, amikor szükséges a többlet teljesítmény.

4.1.3 Burkoló leválasztása és visszaállítása

Forrás:[10]

Az EER (Envelope Elimination and Restoration) módszer zárt hurkú változatát mutatja be a következő ábra. Az RF jel amplitúdó információja leválasztásra kerül egy limiter áramkör segítségével és a szükséges késleltetés után az erősítő bemenetére már csak a fázis által hordozott információ kerül. Az amplitúdó információt burkoló detektorral nyerjük ki, és az erősítő tápfeszültségének változtatásával állítjuk vissza. Ehhez a linearizációs módszerhez szükség van egy jó hatásfokú, gyors reagálású kapcsoló üzemű tápegységre, ami DC feszültséget biztosít a PAM számára. A visszacsatolás segíti a bemeneti amplitúdó követését, ezzel növelve a rendszer

linearitását. A módszer hátránya, hogy kis dinamika tartománnyal rendelkezik, tipikusan 20-30 dB.

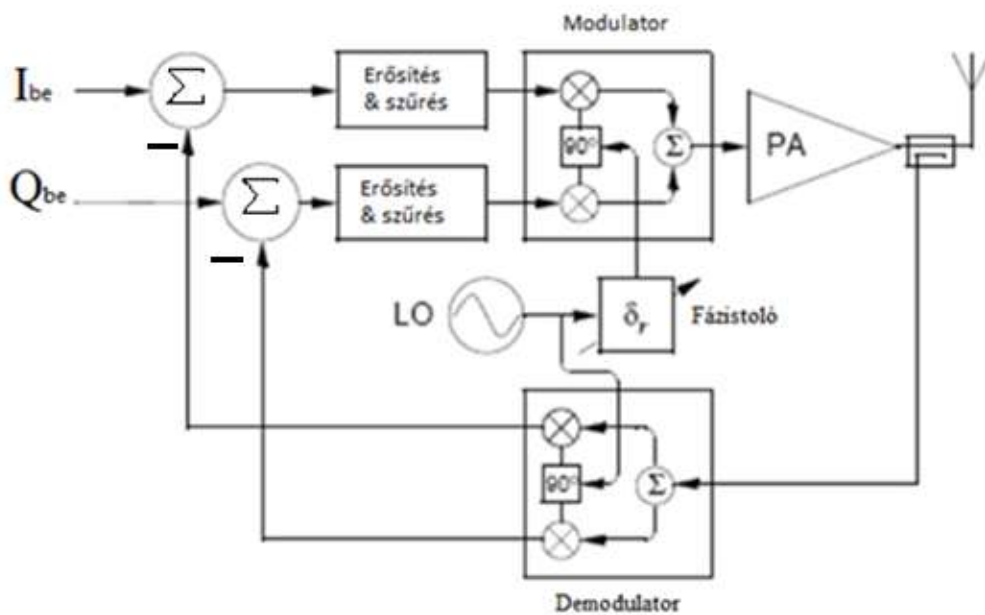


4.2. ábra: Zárthurkú EER.

4.1.4 Cartesian visszacsatolás

Forrás:[10]

Angol nyelvterületen Cartesian feedback-ként ismeri a szakirodalom. A módszer lényege, hogy az alapsávi modulált jel I és Q komponenseit felkeverjük RF sávra majd a kimenőjel egy részét visszakeverjük alapsávra, és egy hagyományos negatív visszacsatolással csökkentjük az erősítő nemlinearitását.



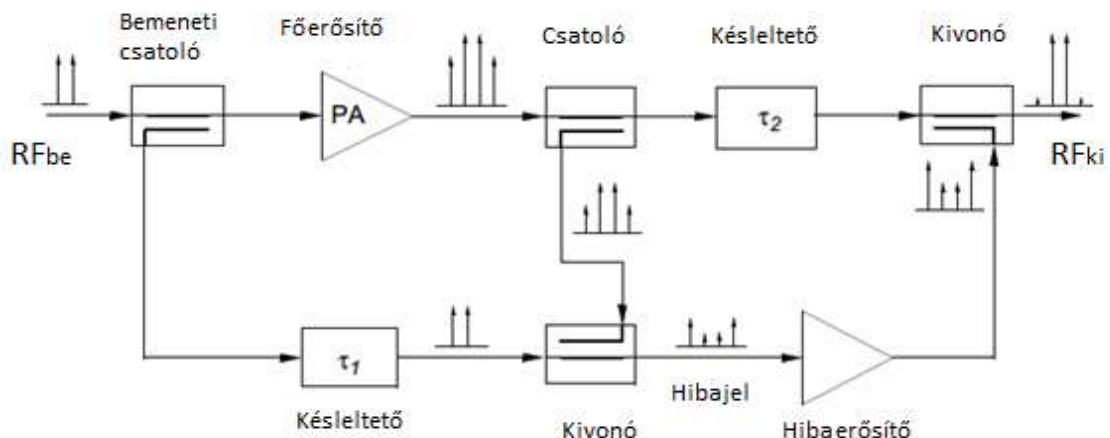
4.3. ábra: Cartesian visszacsatolás.

A visszacsatolóág megkapja a kimenő jel egy részét és a főággal szinkronban demodulálja azt, hogy a visszacsatolás után végrehajtsa a linearizációt. Az intermodulációs torzítás csökkentése egyenesen arányos a főág erősítésével. A rendszert lényegesen könnyebb stabil állapotban tartani, mint ha RF sávon történne a visszacsatolás. Az RF vonalak útkülönbséget muszáj pontosan kikompenzálni a lokál jel egyik ágában lévő fázistolóval, így őrizhető meg a ki- és bemenetek fázisviszonya.

4.1.5 Előrecsatolásos linearizálás

Forrás:[9]

Ez a módszer hullámok destruktív interferenciáján alapszik.



4.4. ábra: Előrecsatolásos linearizálás.

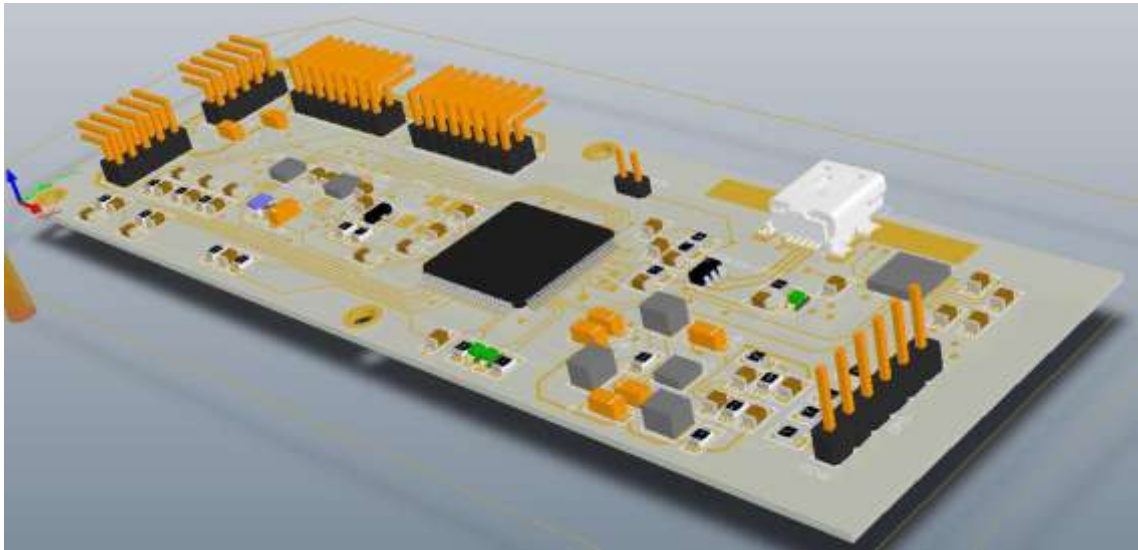
A bejövőjelet a főágban felerősítjük, majd egy részét kicsatoljuk. Az eredeti bejövőjel és a felerősített eltorzult jel különbségét képezzük, megfelelő késleltetés után egy iránycsatolóval. Így kapjuk a hibajelet, ami csak a torzítás során keletkező jelkomponenseket tartalmazza. Ezt felerősítjük egy olyan erősítővel melyet szigorúan a lineáris tartományában használunk. Ez egyszerűen megoldható, hiszen a hibajel által szállított jelteljesítmény több nagyságrenddel kisebb a bejövőjel teljesítményénél. A kimenőjelet a torz jel és a vele azonos szintre hozott hiba jel különbségként kapjuk.

Ilyen módon lehetséges a létező leglineárisabb erősítőket elkészíteni, de a módszernek hátrányai is vannak, mindenből kettő kell. Ez az alkatrész költséget és a termék méretét is növeli. A fázistolók és csillapítók vezérléséhez szükség van egy mikrokontrollerre és vezérlési táblázatokra (a táblázatok felvétele automatizált

méréseket igényel és bonyolult vezérlő szoftvert). Teljesen adaptív megoldás esetén keverőkre, pilotjeleket előállító szintézerre esetleg DSP-re és időközönkénti önkalibrációra van szükség.

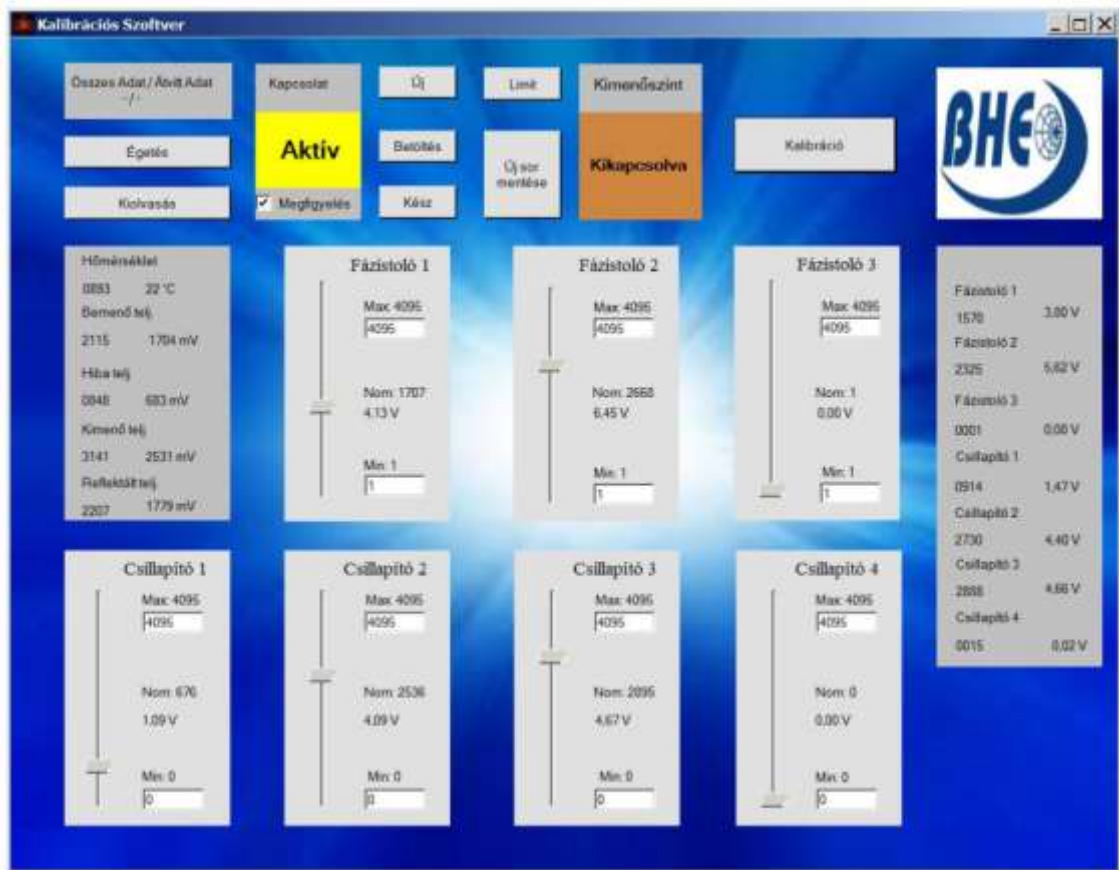
Egy korábbi munkám során már dolgoztam ilyen típusú erősítő fejlesztésén. A projekt célja egy 4Watt kimenőteljesítményű lineáris erősítő előállítása volt. Az én feladatomban a vezérlő szoftver írása, vezérlő elektronika tervezése és mérésautomatizálás volt.

A vezérlő elektronika lelke egy STM32F4 mikrokontroller. Ez a chip képes kezelni a külső DA konvertereket, teljesítmény- és hőmérők analóg kimeneteit mérni illetve kapcsolatot tartani USB-n keresztül a PC-n futó kalibrációs szoftverrel.



4.4. ábra: A vezérlő elektronika.

A grafikus szoftvert C# nyelven írtam. Ez egy többfunkciós program. Elsősorban a mikrokontrollernek küld vezérlő utasításokat melyekkel a felhasználó képes kézzel állítani az analóg beavatkozó tagokat. Ha sikerül egy jó beállítást találni, az eltorzult bejövőjel intermodulációs komponensei és a felerősített hibajel kioltják egymást. Az eszköz kívülről lineárisnak tűnik. A szoftver képes automata módon kalibrációs táblákat készíteni az erősítőkhöz. Ezt a funkciót egy rekúrzív globális minimumhely keresési algoritmusként implementáltam. További hasznos funkció az automata dokumentáció, mellyel gombnyomásra lehet különböző gerjesztésekre spektrumképeket és marker értékeket kimenteni .docx file-okba.

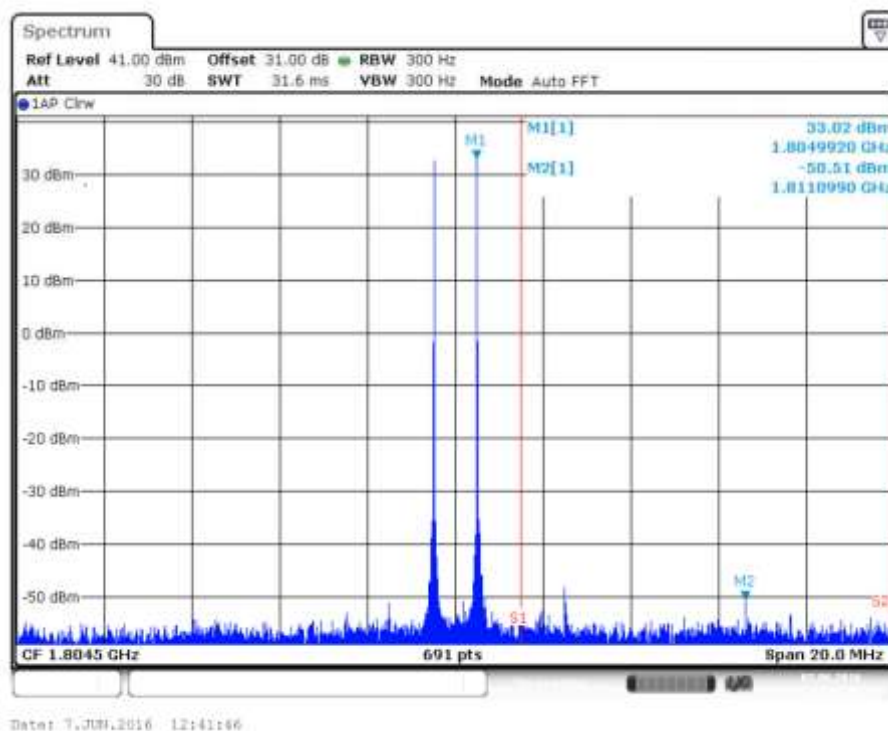


4.5. ábra: A kalibrációs szoftver.

Ahogy már korábban említettem, a gyakorlatban az ilyen erősítők egy mikrokontrollerben eltárolt táblázatból állítják analóg beavatkozótárgaik csillapítását és fázistolását. A táblázatból történő vezérlés azonban nem képes önmagában konzisztensen tökéletes linearizációt elérni. Egy adott időpontban felvett táblázat mely több hőmérsékleten, több teljesítmény értékre is tárolja az egyszer optimálisnak talált beállításokat néhány óra, vagy nap múlva már pontatlan intermoduláció kioltást okoz csak. A végső áttörést egy részleges adaptív algoritmussal értem el. A hibajel teljesítményét méri a mikrokontroller. Ennek a hibajel detektornak az értékétől tettem függővé a főág amplitúdó és fázis korrektorainak beállítását, így folyamatosan minimumon tartva a hibajel teljesítményt. Azért lehetséges ilyen módon megtalálni az optimális beállításokat, mert a hibajel teljesítménye akkor minimális, ha nem tartalmazza a bejövőjelet (tehát csak a torzítás során keletkező jelkomponensekből áll).

A 4.6.-os ábrán látható az eszköz extrém linearitása teljes kivezérlés esetén.

Több, mint 80dBc a különbség a spektrumot szennyező intermodulációs termékek és a jel között. Hagyományos erősítőkkel ilyen jeltisztaságot nem lehet elérni csak az eszköz irracionális túlméretezése árán.

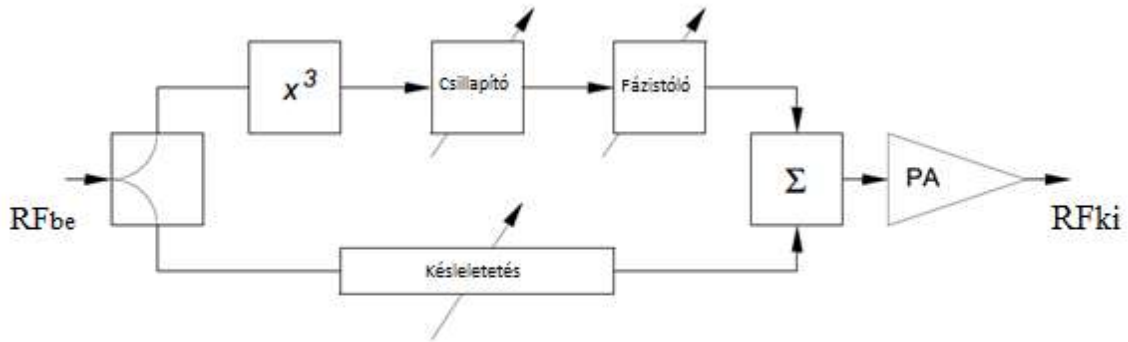


4.6. ábra: Linearizált erősítő teljes kivezérléskor.

4.1.6 Analóg előtorzítás

Forrás:[9]

Az erősítő teljes modellezése nem lehetséges analóg módszerekkel, ahhoz bonyolult jelfeldolgozási algoritmusokra van szükség. Az analóg előtorzítás arra az alapgondolatra épít, hogy a legnagyobb torzítást a harmadrendű intermodulációs termékek okozzák. Ha bejövő jelhez hozzáadjuk annak harmadrendben előtorzított és megfelelő fázisú, amplitúdójú változatát, akkor le lehetne csökkenteni az erősítő eredő torzítását.



4.7. ábra: Analóg előtorzítás.

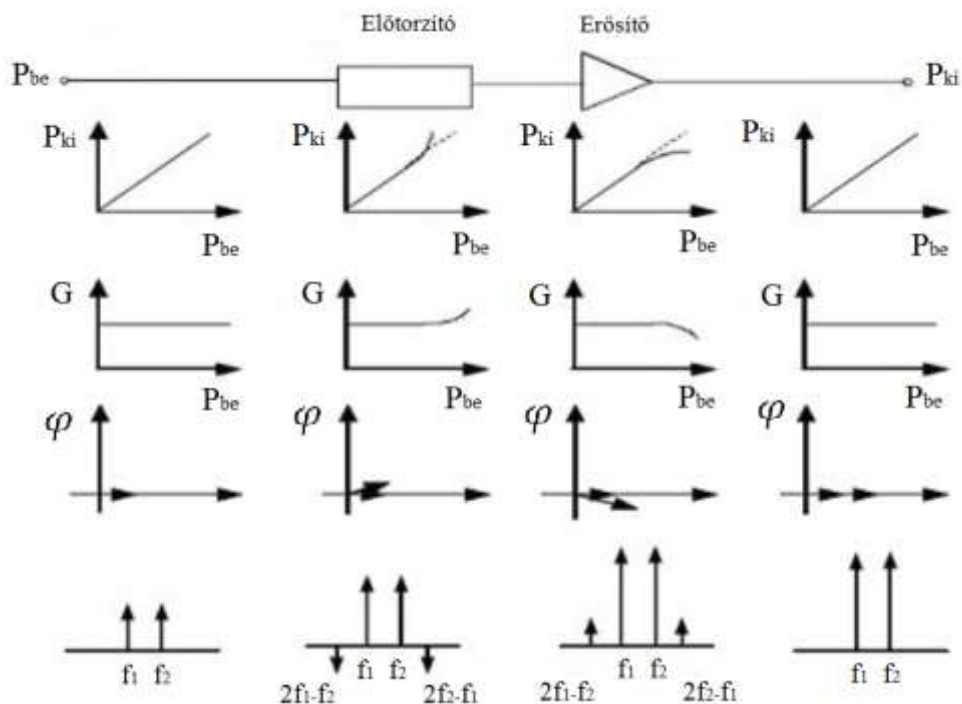
Adaptív módszerekkel lehetséges a hőmérséklet változását és a félvezetők öregedését kompenzálni. A harmadrendű torzítás többféleképpen realizálható. Általában diódával és hibrid iránycsatolóval vagy cirkulátorral oldják meg, de szaturációba vitt erősítővel is lehetséges.

A technika előnye a nagy sáv szélesség, de csak gyengén nemlineáris erősítők esetén hoz kielégítő eredményt. Egy erősen nemlineáris erősítőnél nem csak a harmad, hanem a magasabb rendű komponenseket is ki kell oltani.

4.2 Digitális előtorzítók

Forrás:[4]

Ezek a linearizálási módszerek arra épülnek, hogy ha elő tudjuk állítani az erősítőnek, mint rendszernek az inverz modelljét, akkor olyan módon tudjuk eltorzítani a jelet, hogy az áthaladva a nemlineáris erősítőn pontosan az előtorzító bemenetére adott jel G_0 -szorososa legyen.



4.8. ábra: Előtorzítás.

Egy előtorzító igen nagymértékben növeli meg az erősítő bonyolultságát. A bejövő RF jelet le kell keverni alap, vagy közép sávra, mintavételezni, majd a jelfeldolgozás után újra felkeverni. Csak azokban az alkalmazásokban van létjogosultsága, ahol a mikrohullámú áramkörü részek költsége nagy. Ilyenek a bázisállomások, repeaterek és műholdak transzponderei.

A digitális előtorzítókat sokféleképpen lehet kategorizálni, ahogy azt a 4.9. ábra is mutatja.



4.9. ábra: A digitális előtorzítók felosztása.

4.2.1 A jelfeldolgozó láncban betöltött hely szerinti felosztás

Megkülönböztetünk adat és jel előtorzítókat. Az **adat előtorzítók** a moduláció konstellációs pontjait változtatják meg úgy, hogy az erősítő nemlinearitása a megfelelő „helyre” rakja a pontokat. Az adat előtorzítás nem képes kiküszöbölni a szomszédos csatornába kerülő szivárgó teljesítmény problémáját. Képesek akár a teljes csatorna torzítását kikompenzálni egy vevő és adó áramkör között. Legnagyobb hátrányuk, hogy működésük függ a moduláció konstellációs diagramjától.

A **jel előtorzítók** már a modulált jelet módosítják. Előnyük, hogy működésük kevésbé függ a moduláció típusától és a vivők számától.

4.2.2 Memória szerinti felosztás

Vannak olyan előtorzítók, melyek az előtorzításhoz figyelembe veszik az erősítők memóriahatását és vannak olyanok, amik nem. A **memóriamentes** előtorzítók lényegesen egyszerűbbek. Olyankor érdemes használni őket, amikor a teljesítményerősítő jól modellezhető memóriamentes modellel is. Ilyen eset az, amikor keskenysávú jelet akarunk erősíteni, vagy a kimenő szint nem annyira nagy, hogy érdemben megváltoztassa a DC csatoló hálózat impedancia viszonyait.

Memóriával rendelkező előtorzítók a memória mélységével arányosan növelik a számítás igényt, de a modern szélessávú kommunikációs rendszerekben használatuk imperatív. Olyankor is szükség lehet használatukra, amikor a teljesítmény FET periodikus felmelegedése és lehűlése visz memóriát a rendszerbe.

4.2.3 Adaptáció szerinti felosztás

Léteznek adaptív és nem adaptív előtorzítók. Az **adaptív előtorzítók** előnye, hogy képesek a különböző működési feltételekhez (sávszélesség, hőmérséklet, teljesítmény) alkalmazkodni. Képesek továbbá az erősítő öregedéséből adódó változásokat teljes élettartamuk alatt kompenzálni. Ezek értékes képességek olyan eszközök esetén melyek karbantartása irracionálisan drága vagy műszakilag kivitelezhetetlen (pl. műholdak).

A **nem adaptív** előtorzítók előnye az egyszerűbb hardver és szoftver architektúra, és az ezekből adódó alacsonyabb előállítási költség. Az adaptív előtorzítókkal szemben nem kell bizonyos időközönként újra kalibrálniuk magukat így üzemeltetésüket nem kell megszakítani. A változó működési feltételekhez előre felvett kalibrációs táblákat használnak.

4.2.4 Az előtorzítás kivitelezése szerinti felosztás

Kétféleképpen lehet kivitelezni magát az előtorzítás műveletét. A **parametrikus** előtorzítóknál egy nemlineáris modellt (3. fejezet) alkotunk, mely a jelfeldolgozó áramkörön implementálásra kerül. Ennek a modellnek a megalkotása LS vagy iteratív módszerrel történhet (4. fejezet). A modellben nyilvánvalóan szükség lesz a bejövő jel hatványozására és konstansokkal való szorzásra. Ezek számításigényes műveletek, melyek korlátozzák az alkalmazható modell bonyolultságát és az egységnyi idő alatt érkező minták számát (tehát a jel sávszélességét).

A **LUT alapú** előtorzítás használatakor az erősítő inverz karakterisztikáját lineáris egyenesekkel közelítjük (nem kötelező) és egy táblázatból keressük ki, hogy adott bejövő jelhez mekkora kimenő jel tartozik. Érdekes kérdés, hogy milyen módon osztjuk el a LUT bejegyzéseket (nagyobb teljesítményeken hasznos lehet több bejegyzést készíteni) és hogyan interpolálunk köztük. A LUT alapú eszközök a gyakorlatban néhány dB-el alacsonyabb ACPR-t tudnak elérni a parametrikusaknál. Ilyenkor a parametrikus számítási problémát (szorzás, hatványozás) keresési problémává

transzformáljuk. Ettől kezdve a táblázatainkban lefolytatott keresés gyorsasága fog felső határt szabni a sávszélességnek és modell komplexitásnak.

5. Matematikai összefoglaló

Forrás:[11]

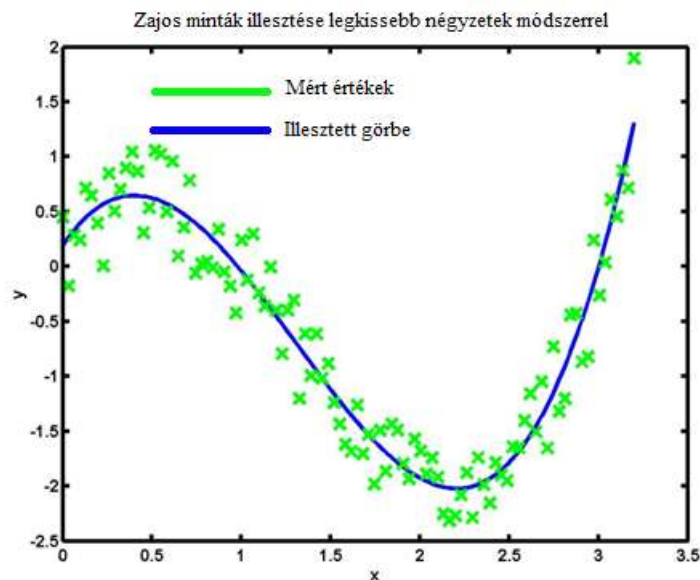
Ebben a fejezetben bemutatásra kerülnek azok a matematikai módszerek és algoritmusok, melyek szükségesek rendszerek identifikálásához.

5.1 Legkisebb négyzetek módszere

Forrás:[12]

A legkisebb négyzetek módszere jól bevált megoldás a regresszió analízis területén. Segítségével túlhatározott egyenletrendszerek közelítő megoldását kaphatjuk meg. A „legkisebb négyzetek” azt jelenti, hogy a megoldás minimalizálja az egyes egyenletek megoldásaiból származó hibák négyzetének összegét. Alkalmazásának legfontosabb területe a görbe illesztés. A mért adatokra olyan görbét illeszt mely négyzetes értelemben a legkevésbé tér el a mért értékektől.

Forrás:[13]



5.1. ábra: Legkisebb négyzetek módszere.

5.2 Polinomiális regresszió

A polinomiális regresszió értelmezéséhez először tekintsük át egyenletrendszerek optimális megoldását.

Ha egy ismeretlen mennyiség meghatározására méréseket végzünk, általában túlhatározott egyenletrendszert kapunk. Több egyenletünk lesz mint változónk, és a mérési pontatlanságok miatt nem lesz konzisztens (megoldható) az egyenletrendszer. Célszerű tehát az alábbi egyenletrendszer pontos megoldása helyett:

$$\mathbf{X}\mathbf{a} = \mathbf{y} \quad (5.1)$$

az \mathbf{a} vektor optimális megoldását keresni, mely legkisebb négyzetes értelemben a lehető legkevésbé tér el a többi lehetséges megoldásvektortól.

$$\mathbf{X}^T\mathbf{X} \hat{\mathbf{a}} = \mathbf{X}^T\mathbf{y} \quad (5.2)$$

Az 5.2-es egyenlet megoldásai megegyeznek az 5.1-es egyenlet optimális megoldásaival. Ezek közül egyetlen egy esik az \mathbf{X} mátrix sorterébe, a legkisebb abszolút értékű.

Tekintsük a következő, k -ad rendű polinomokból álló egyenletrendszert:

$$\begin{aligned} \mathbf{a}_0 + \mathbf{a}_1\mathbf{x}_1 + \dots + \mathbf{a}_k\mathbf{x}_1^k &= \mathbf{y}_1 \\ \mathbf{a}_0 + \mathbf{a}_1\mathbf{x}_2 + \dots + \mathbf{a}_k\mathbf{x}_2^k &= \mathbf{y}_2 \\ \vdots & \\ \mathbf{a}_0 + \mathbf{a}_1\mathbf{x}_n + \dots + \mathbf{a}_k\mathbf{x}_n^k &= \mathbf{y}_n \end{aligned} \quad (5.3)$$

Amennyiben az együttható mátrixot \mathbf{X} jelöli, az ismeretlenek vektorát \mathbf{a} , az y_i mért értékek vektorát \mathbf{y} akkor az egyenletrendszer $\mathbf{X}\mathbf{a} = \mathbf{y}$ alakba írható. Az optimális megoldás:

$$\mathbf{X}^T\mathbf{X} \hat{\mathbf{a}} = \mathbf{X}^T\mathbf{y} \quad (5.4)$$

$$\hat{\mathbf{a}} = (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{y} \quad (5.5)$$

Ebből már látható, hogy polinomokkal modellezett rendszereket polinomiális regresszióval identifikálhatunk. Azonban $\hat{\mathbf{a}}$ előállításához $\mathbf{X}^T\mathbf{X}$ mátrix inverzével balról meg kell szoroznunk a kifejezést. Ez az inverz nem feltétlenül létezik.

5.3 Pszeudóinverz

Legyen \mathbf{X} egy $n \times n$ -es mátrix. \mathbf{X} -et invertálhatónak mondjuk, ha létezik egy \mathbf{Z} mátrix melyre igaz, hogy $\mathbf{XZ}=\mathbf{ZX}=\mathbf{I}$. \mathbf{Z} mátrixot \mathbf{X} inverzének nevezzük és \mathbf{X}^{-1} -vel jelöljük.

Négyzetes együtthatómátrixú, konzisztens egyenletrendszerek esetén használható a mátrix inverzével balról történő szorzás. Ilyenkor az egyenletrendszer pontos megoldását kapjuk. Ha azonban \mathbf{X} mátrix nem négyzetes vagy nem konzisztens, akkor szükséges az inverz fogalmát általánosítanunk. Ha gyengítünk a feltételünkön és megelégszünk az optimális megoldással, akkor bevezethetjük a pszeudóinverz fogalmát.

A pszeudóinverzen olyan mátrixot értünk, amely képes tetszőleges $\mathbf{X}\mathbf{a} = \mathbf{y}$ egyenletrendszer minimális abszolút értékű, $\hat{\mathbf{a}}$ optimális megoldását (5.1. fejezet) megadni. Tehát \mathbf{X}^+ -al jelölve a pszeudóinverzet: $\hat{\mathbf{a}} = \mathbf{X}^+\mathbf{y}$.

5.4 NLMS algoritmus

Forrás: [14]

A Least Mean Squares algoritmus is modellillesztésre használható. Nem igényli a modellezendő rendszer statisztikai jellemzőinek ismeretét, hanem iteratív módon közelíti az optimális paramétereket. Az, hogy nem igényli a rendszer statisztikai tulajdonságainak ismeretét nagy jelentőséggel bír, mert ezek a paraméterek a gyakorlatban ritkán állnak rendelkezésre. Minden iterációban a rendszer a pillanatnyi hibával ellentétes irányban, de arányosan változtatja a paraméter vektor elemeit. Legyen \mathbf{x} vektor a modell bemenete, melyet \mathbf{a} vektorral súlyozunk. A kimenetünk y skalár, mely e hibával tér el a megkövetelt d kimenettől. Az n . iterációban így adódik a következő súly vektor:

$$\mathbf{a}_{n+1} = \mathbf{a}_n + \mu e_n \mathbf{x}_n \quad (5.6)$$

μ a bátorsági tényező, melynek értékét általában empirikus módon, az adott alkalmazástól függően kell megválasztani. Nagy μ esetén gyorsabban konvergál az

algoritmus az optimális súlyvektorhoz, de instabillá is válhat a rendszer. Kis μ esetén lassabb a konvergencia, de stabilabb és pontosabb végeredményt ad. μ -t érdemes viszonylag nagy értékről indítani, majd lépcsőzetesen csökkenteni.

A pillanatnyi hibát a következő összefüggéssel kaphatjuk meg:

$$e_n = d_n - y_n = d_n - \mathbf{a}_n^T \mathbf{x}_n \quad (5.7)$$

Az \mathbf{a} vektor tartalmazza azokat a modell paramétereket melyek közelítik az optimális együttható készletet.

A fenti egyenletek a klasszikus LMS algoritmust írják le. A Normalized Least Mean Squares módszer (NLMS) azonban a bemeneti vektor nagyságával normálja a bátorsági tényezőt így biztosítva a konvergenciát. Mivel az együtthatók konvergenciája függetlenné válik a bemenet nagyságától nagyobb μ is használható, mely gyorsítja az identifikációt. α -t regularizációs konstansnak hívják, nulla bemenet esetén megakadályozza a nullával való osztást.

$$\mathbf{a}_{n+1} = \mathbf{a}_n + \frac{\mathbf{x}_n \mu e_n}{\alpha + \mathbf{x}_n^T \mathbf{x}_n} \quad (5.8)$$

5.5 Kereszt korrelációs időkülönbség korrekció

Forrás: [19]

A jelfeldolgozásban a keresztkorreláció két mintasor hasonlóságának mértéke a köztük lévő időkülönbség függvényében.

Folytonos függvények esetén (a^* komplex konjugáltat jelöli):

$$(f * g)(\tau) \stackrel{\text{def}}{=} \int_{-\infty}^{+\infty} f^*(t)g(t + \tau)dt \quad (5.9)$$

Hasonlóan diszkrét függvények esetén:

$$(f * g)[n] \stackrel{\text{def}}{=} \int_{-\infty}^{+\infty} f^*[n]g[n + m]dt \quad (5.10)$$

Tegyük fel, hogy: $g[n] = S\{f[n-M]\}$ ahol S valamilyen lineáris vagy nemlineáris memóriával rendelkező rendszer hatása a bemenetére adott f jelnek. Ha meg szeretnénk határozni M értékét, a rendszer késleltetését, akkor ki kell számolnunk különböző m -ekre a két jel kereszt korrelációját és az az m , amire a kereszt korreláció maximális (vagy minimális negatívan korrelált jelek esetén), az fog megfelelni M -nek. Tehát a következő szélsőérték keresést kell elvégezni a mintákon:

$$M = \operatorname{argmax} ((f * g)[m]) \mid m \quad (5.11)$$

6. Előtorzítók paraméter becslése

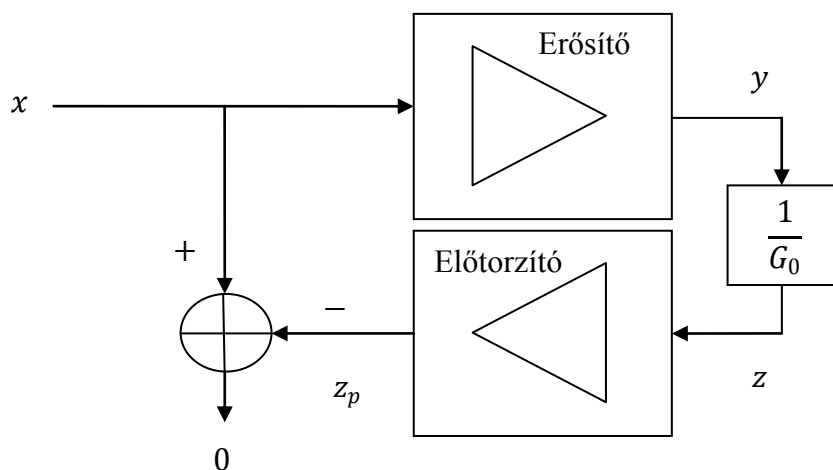
Két módszer terjedt el előtorzítók paramétereinek meghatározására. Ezek a módszerek erősen összefüggnek a használni kívánt modellalkotás matematikai apparátusával.

6.1 Inverz tanuló architektúra

Forrás:[7]

Azért hívják inverz architektúrának, mert az erősítő kimeneti jelét használja az előtorzító paraméterbecslésének bemeneteként. A polinomiális regresszió (5.2. fejezet) módszerével mérési eredményekből meghatároz egy polinomot, mely közelítőleg megadja a modellezni kívánt nemlineáris függvénykapcsolat inverzét. A modellezendő függvénykapcsolat pedig nem más, mint az erősítő nemlineáris karakterisztikája.

Ennél az architektúránál különbözik a „tanító” és a működés közben használt topológia. Az Inverse Learning Architecture (ILA) egy lépésben határozza meg az előtorzító együtthatóit N darab mérési eredmény felhasználásával. A bemeneti jel x áthalad a teljesítmény erősítőn és eltorzul, így kapjuk y -t. Az tanuló előtorzító bemenete: $z = \frac{y}{G_0}$.



6.1. ábra: Inverz tanuló architektúra tanulási konfigurációja.

Tekintsük át a linearizálási architektúra alkalmazásának részleteit:

A tanuló architektúra ábrájának elnevezéseit használva a következő egyenletrendszert írhatjuk fel, ahol \mathbf{Z} az előtorzító bemenetéből képzett együttható mátrix, az \mathbf{a} a keresett paraméter vektor, \mathbf{x} pedig az elvárt kimenet:

$$\mathbf{Z}_{N \times KL} \mathbf{a}_{KL} = \mathbf{x}_N \quad (6.1)$$

Az előtorzító egy K minta hosszú memóriahatást és L -ed rendű nemlinearitást kompenzálni tudó memória polinom.

$$x_{[n]} = z_{p[n]} = \sum_{k=0}^{K-1} \sum_{l=0}^{L-1} a_{kl} \Phi_{kl} = \sum_{k=0}^{K-1} \sum_{l=0}^{L-1} a_{kl} z_{[n-k]} |z_{[n-k]}|^l \quad (6.2)$$

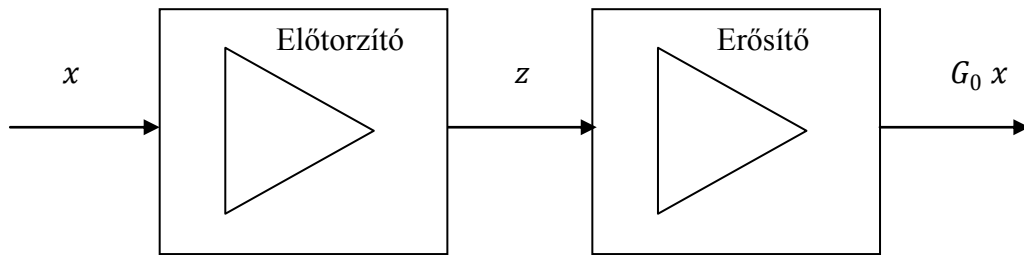
Tehát a \mathbf{Z} mátrix N sorból (mérési eredményből) és $K \cdot L$ darab oszlopból áll.

$$\mathbf{Z} = \begin{matrix} \Phi_{00}[0] & \dots & \Phi_{0L-1}[0] & \dots & \Phi_{k0}[0] & \dots & \Phi_{K-10}[0] & \dots & \Phi_{K-1L-1}[0] \\ \Phi_{00}[1] & \dots & \Phi_{0L-1}[1] & \dots & \Phi_{k0}[1] & \dots & \Phi_{K-10}[1] & \dots & \Phi_{K-1L-1}[1] \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \Phi_{00}[N-1] & \dots & \Phi_{0L-1}[N-1] & \dots & \Phi_{kl}[N-1] & \dots & \Phi_{K-10}[N-1] & \dots & \Phi_{K-1L-1}[N-1] \end{matrix}$$

A keresett \mathbf{a} vektor az \mathbf{x} vektor \mathbf{Z} mátrix pszeudó inverzével (5.3) balról történő szorzással kapható:

$$\mathbf{a}_{KL} = \mathbf{Z}_{KL \times N}^+ \mathbf{x}_N \quad (6.3)$$

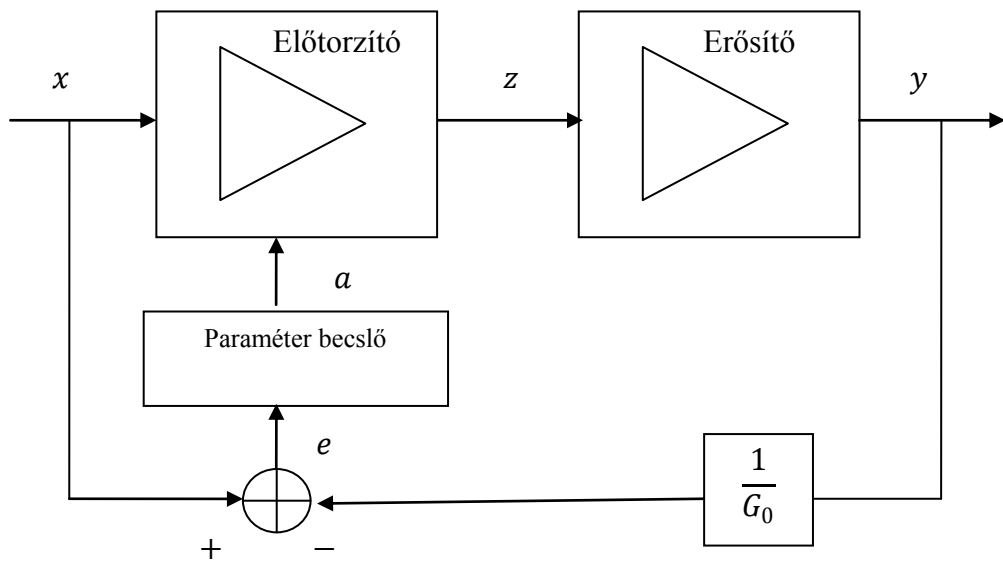
Mivel $f(f^{-1}x(t)) = f^{-1}(f(x(t)))$ ezért rendes működéskor (6.2.-es ábra) a DPD blokkot az erősítő elé helyezve a linearizált, eredő erősítőkarakterisztikát kapjuk. Ha az előtorzító „megtanulta”, hogy a torz y/G_0 bemenő jelre pontosan x jelet kell előállítania, (6.2) akkor torzítatlan x jelre olyan z jelet fog produkálni, hogy az áthaladva a PAM-on a bemenet G_0 -szorososa lesz. Tehát az erősítés lineárisává válik.



6.2. ábra: Inverz tanuló architektúra működési konfigurációja.

6.2 Direkt tanuló architektúra

Ez a módszer az a vektor iteratív approximációján alapszik. Ha sikerül megtalálni az optimális együtthatók vektorát, akkor a torzítás minimális lesz.



6.3. ábra: Direkt tanuló architektúra.

Az előtorzító egy L -ed rendű nemlinearitást kompenzálni tudó polinom.

$$z_{[n]} = \sum_{l=0}^{L-1} a_l \Phi_l = \sum_{l=0}^{L-1} a_l x_{[n]} |x_{[n]}|^l \quad (6.4)$$

Az \mathbf{a} vektor értékét minden lépésben az NLMS algoritmus szerint frissítjük (5.4. fejezet), amíg nem csökken kellően kicsire a hiba (e). Ideális esetben:

$$y = G_0 x. \tag{6.5}$$

7. Szimulációk

Az előző fejezetekben bemutatott elméleti ismereteket felhasználva Matlab szimulációkat készítettem. A szimulációk célja az volt, hogy kipróbáljam a fent részletezett modellalkotási módszereket, algoritmusokat.

A szimulációkhoz szükség volt szélessávú gerjesztő jelre, nemlinearitással és memóriával rendelkező erősítő modellre, előtorzítóra és a jelek frekvenciatartománybeli ábrázolására.

Gerjesztés:

Az erősítő identifikálását szélessávú, modulált jellel szokás végezni. Mivel nem tartottam szükségesnek saját modulátort is írni Matlabban, ezért változtatható alapsávi sáv szélességű multiszínuszt használtam. A jelkomponensek száma tetszőlegesen változtatható. Az egyes szinuszok véletlen kezdőfázisúak, így minden futáskor más időfüggvényű jellel dolgozhattam. Az összesített jelteljesítmény 50 Ohmos környezetet feltételezve, dBm-ben megadható.

Erősítő:

Az erősítőt egy Wiener modellel definiáltam, melynek paramétereit empirikus módon határoztam meg.

Előtorzító:

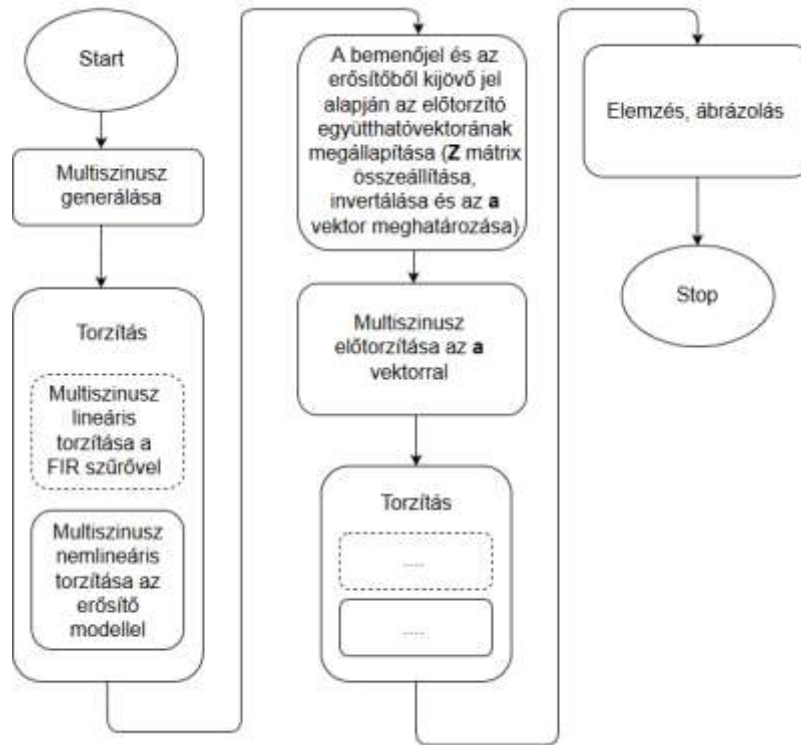
Memória Polinomot (3.2.4. fejezet) használtam az előtorzítók megalkotására. Elsőnek az inverz tanuló architektúrát próbáltam ki úgy, hogy az erősítő memória hatását kikapcsoltam. Az erősítő így nem mutat semmilyen frekvencia szelektív-viselkedést

A modulátor kivételével az ábrázolt jeleket kényelmi okokból visszaosztottam a kisjelű erősítéssel.

7.1 Szimulációk Inverz Tanuló Architektúrával

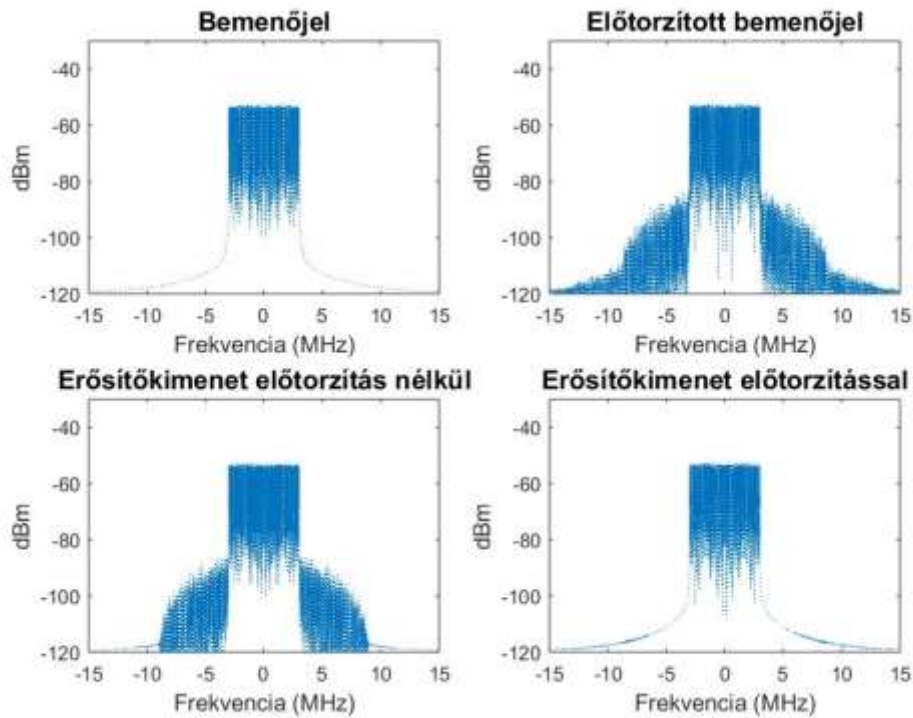
A most következő szimulációk matematikai és modellalkotási kérdéseit a 3.2.4, 5.1, 5.2, 5.3, 6.1 fejezetekben részletesen bemutattam.

A következő folyamatábrán látható, hogy milyen lépésekből áll a szimuláció. A szaggatott vonallal jelölt FIR szűrő ebben az esetben nincs használatban.



7.1. ábra: Az ILA szimuláció folyamatábrája.

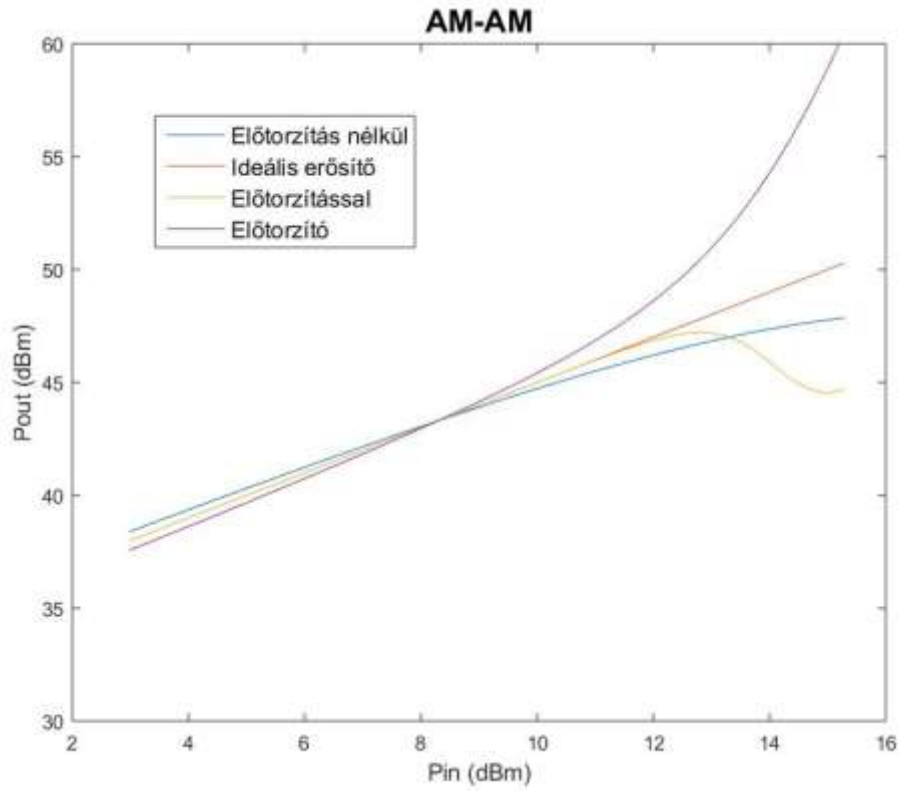
A szimulációk megkezdése előtt be kell állítani, hogy hányadfokú polinommal (L) és hány mintányi memória mélységgel (K) szeretnénk elkészíteni az előtorzítót. Meg kell adni a Z mátrix előállításához használt minták (N) számát. A mintaszám növelése nem vezet pontosabb előtorzító modellhez egy bizonyos szám fölött.



7.2. ábra: ILA szimuláció memóriamentes erősítő esetén.

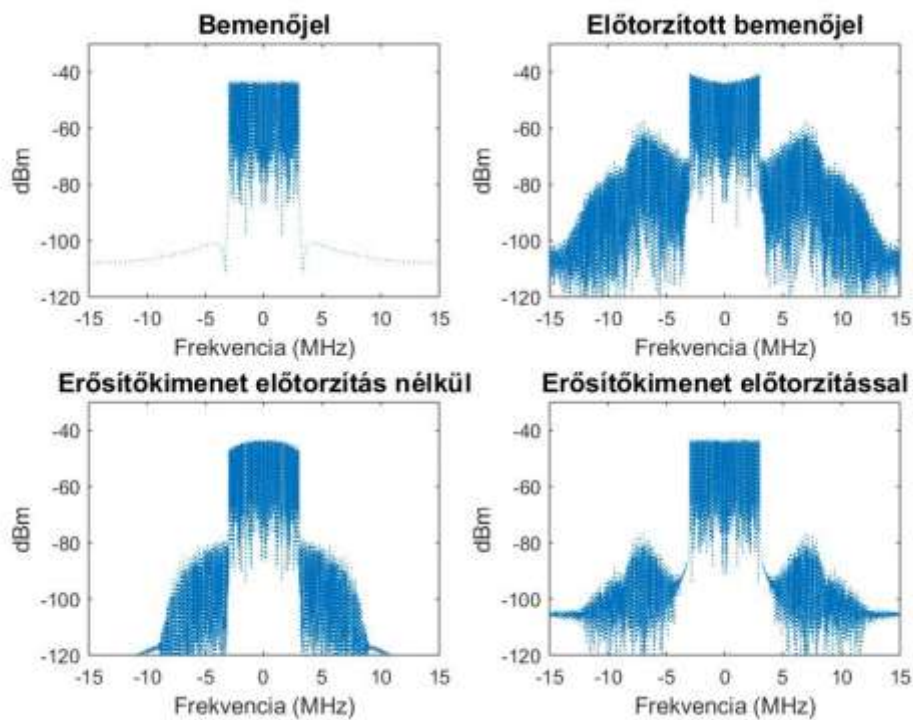
A használt előtorzító ötödrendű, memóriamentes nemlinearitások modellezésére alkalmas. A gerjesztőjel teljesítménye 6dBm. Az intermodulációs torzítás jelentősen csökkent, tehát jól működik az előtorzító.

Tekintsük most az erősítő AM – AM görbét. Az előtorzító karakterisztikáját 35.5dB-lel megemeltem a jobb értelmezhetőség kedvéért. Az előtorzított jel jól követi az ideális egyenest és magasabbra tolja az erősítő 1dB-es kompressziós pontját. A linearizáláshoz fontos ismerni a pontos kisjelű erősítést. Az összes linearizálási algoritmus erősen épít erre az apriori információra.



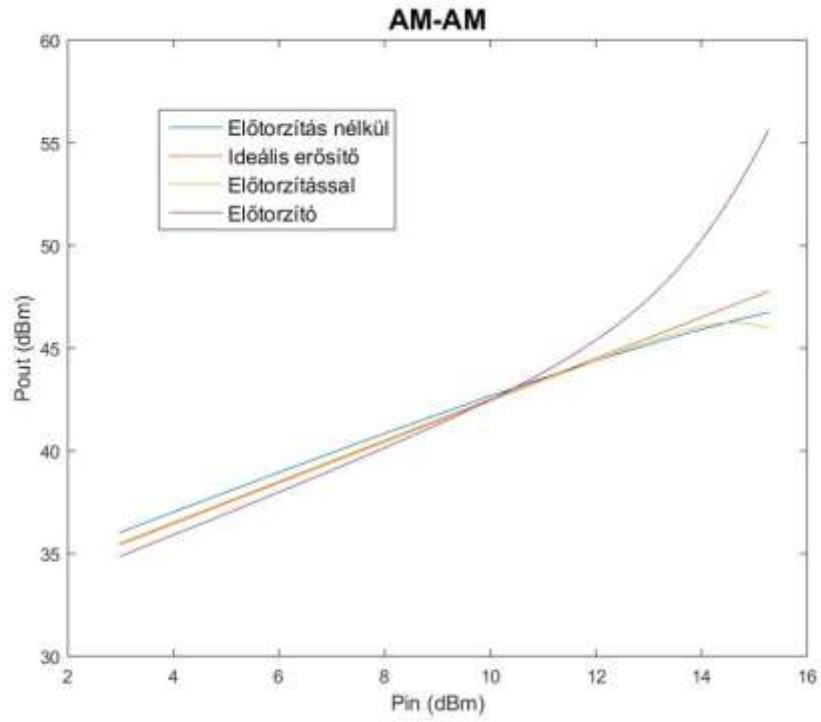
7.23. ábra: Karakterisztikák memóriamentes erősítő esetén.

Újra futtatva a szimulációt úgy, hogy egy aluláteresztő FIR szűrővel modelleztem az erősítő dinamikus részét a következő spektrumokat kaptam:



7.4. ábra: ILA szimuláció memóriával rendelkező erősítő esetén.

Az előtorzító most egy tízmintányi memóriával bír, ötöd rendű nemlinearitással kompenzáló memória polinom. Az erősítő frekvencia-szelektív átvitelét képes kikompenzálni az eszköz, de az intermodulációs termékek elnyomása ilyenkor nehezebbé válik. Míg előző esetben az előtorzításhoz használt vektor hossza öt volt, most már ötven. Ezt az implementáció során majd figyelembe kell venni. A memória mélység növekedésével arányosan nő a szükséges számítási igény is.



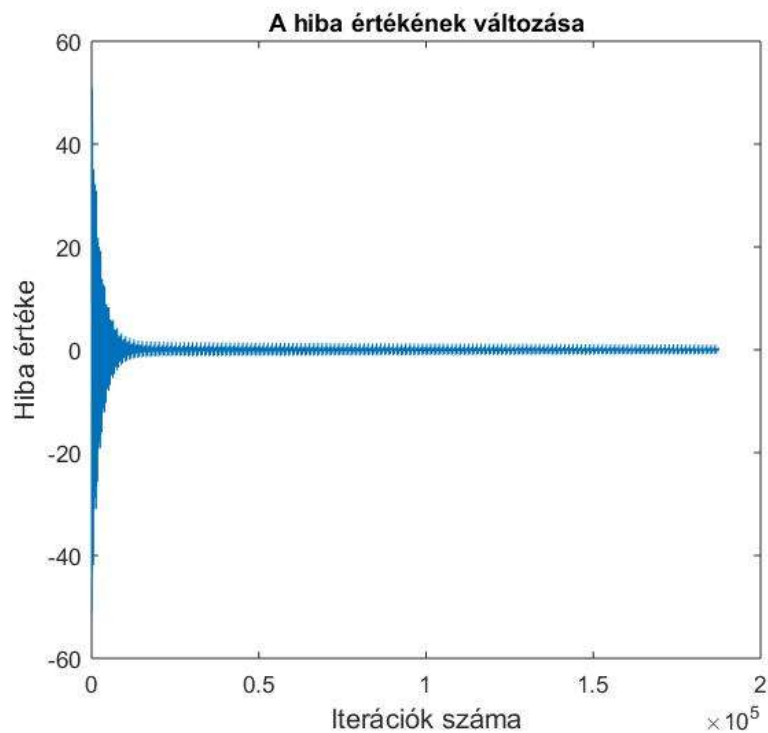
7.5. ábra: Karakterisztikák memóriával rendelkező erősítő esetén.

7.2 Szimulációk Direkt Tanuló Architektúrával

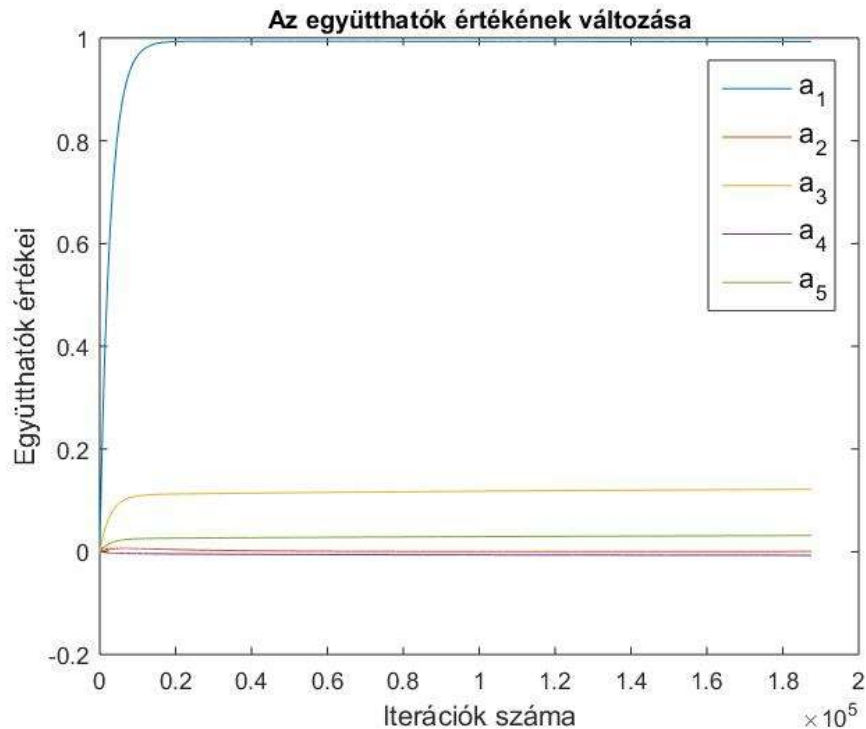
A szimulációk második szakaszában a direkt tanuló architektúrával és az NLMS algoritmussal kísérleteztem. A módszerek részletes ismertetése az 5.4 és 6.2 fejezetekben található. Az NLMS algoritmus önmagában csak késleltetés nélküli, tisztán statikus rendszereket képes kikompenzálni. Ennek megfelelően statikus polinomként modelleztem az előtorzítót (ismételve a 6.4-et):

$$z_{[n]} = \sum_{l=0}^{L-1} a_l \Phi_l = \sum_{l=0}^{L-1} a_l x_{[n]} |x_{[n]}|^l \quad (6.4)$$

Az Fx-NLMS algoritmus képes lenne dinamikus rendszereket identifikálni, de nem tudtam megoldani, hogy egyben nemlinearitásokat is kezeljen. Egy tényleges implementációnál a bemenet mintáinak és a visszacsatolt jel mintáinak időkülönbségét egyéb módszerekkel is lehet korrigálni (Lásd: 5.5).



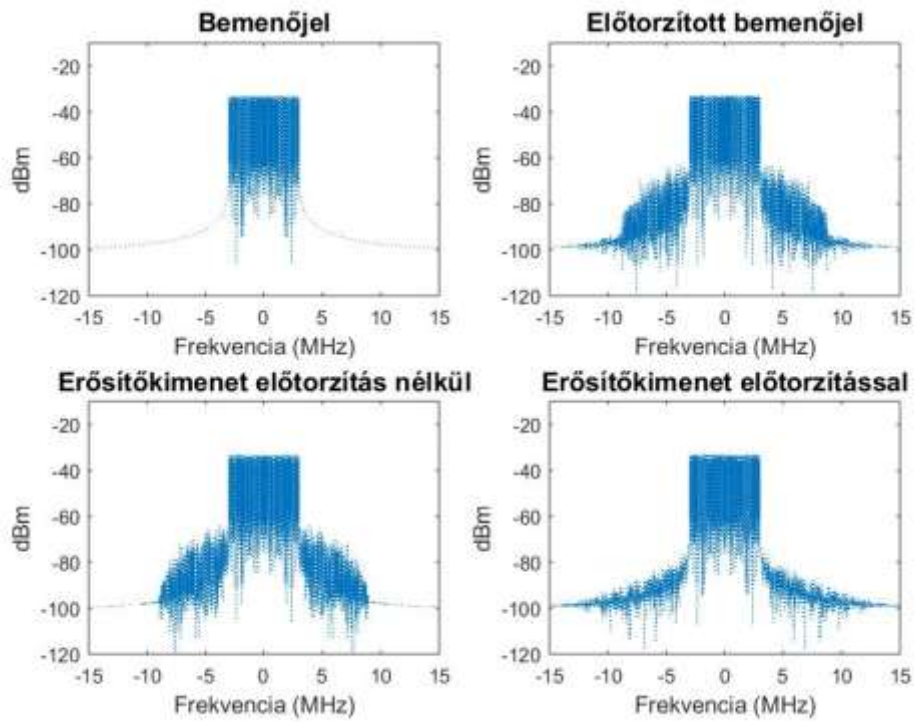
7.6. ábra: A hiba változása iterációnként.



7.7. ábra: Egy ötödrendű előtorzító polinom együtthatói.

A fenti két ábrán látható az előtorzító néhány tulajdonsága. A hiba nem konvergál nullához, de korlátossá válik egy idő után és tapasztalataim szerint kellően kicsi lesz a kielégítő linearizációhoz. A gerjesztő multiszínű jel kvázi periodicitását próbáltam csökkenteni fehér zaj hozzáadásával. Sajnos ez semmiféle javulást nem okozott a hiba értékének csökkenésében illetve ugyan úgy periodikus maradt.

Ahogy a 7.7. ábrán látszik az előtorzító polinomjainak együtthatói konvergenciát mutatnak. A bátorsági tényezőt $0.001 > \mu > 0.00000001$ között kellett tartanom, hogy ne váljon instabillá a rendszer. Érdeemes megjegyezni, hogy csak a páratlan rendű polinomokhoz tartozó együtthatók vannak hatással a kimenetre. Ez egy fontos, de korántsem váratlan eredmény. Az intermodulációs torzítást a nemlineáris rendszerek páratlan fokszámú tagja okozzák. Egy valós implementáció során takarékosabban lehet bánni az erőforrásokkal, ha csak páratlan fokszámú tagjai vannak az előtorzító polinomnak.



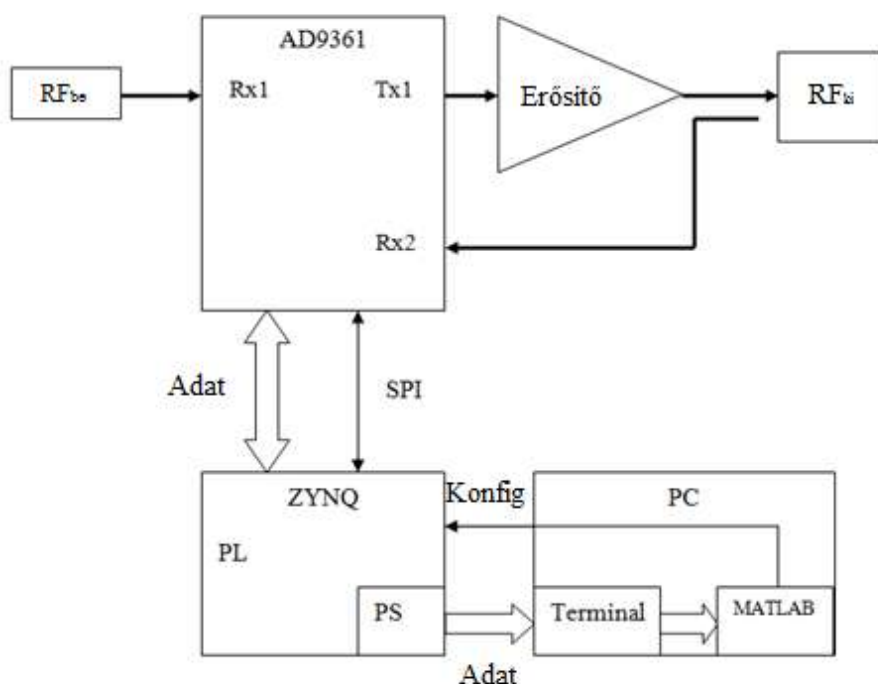
7.8. ábra: Az előtorzító hatása a jel spektrumára .

Ahogy a 7.8-as ábrán látszik az előtorzító sikeresen csökkenti az intermodulációs torzítást, de várhatóan rosszabb ACPR érhető el vele, mint az ILA típusú előtorzítóval.

8. Implementáció

A munka első felében arra a következtetésre jutottam, hogy egy memória polinom alapú inverz tanuló architektúrájú előtorzítót lenne érdemes készíteni. Ennek a fajta előtorzítónak a megvalósításához nem szükséges hozzáférni a jel konstellációs diagramjához a modulátorban, de sokfajta nemlinearitást és memória effektust képes kompenzálni, akár adaptív módon. A megvalósításhoz rendelkezésemre állt egy ZedBoard, DualCore Zynq Z-7020-as SoC-val, illetve egy FCOMMS-3 fejlesztő panel mely tartalmazott egy AD9361 adóvevő chipet.

Az RF panelt kifejezetten arra a célra tervezték, hogy a ZedBoardal együtt használva gyorsan lehessen szoftver rádiós rendszereket kifejleszteni. A következő blokk diagramon látható, hogy milyen jelfeldolgozó hardvert akartam létrehozni:



8.1. ábra: Az implementált rendszer.

Ahogy a fenti ábrán látható, az volt az elképzelésem, hogy az AD9361-es chippet mint egy „vezeték darabot” használom a jel továbbítására. A chip lekeveri, mintavételezi, szűri és továbbítja az FPGA felé a jelet, de nem módosítja azt. Ilyen feltételek mellett szert teszek a gerjesztő jel és a torzított jel mintáira, melyeket át tudok

adni a PC-n futó Matlabnak. Minden magasabb rendű matematikai műveletet így már meglévő scriptekkel tudok elvégezni anélkül, hogy implementálnom kellene őket beágyazott környezetben. Szükséges egy mátrix pszeudó inverzét is kiszámolnom többek közt, ezért ez a módszer jelentős időmegtakarítást jelentett a fejlesztés során. Mivel így rugalmassá vált a fejlesztői környezet, sokkal több mindent tudtam kipróbálni rövid idő alatt. Egy fejlesztési projektnél ez fontos tényező.

A mintavételezett jeleket (külön I és Q komponensek) felhasználva kiszámoltam az erősítő inverzét modellező memória polinomok együtthatóit. Az FPGA blokk RAM-jaiban létrehoztam két LUT-ot melyek rekeszeit a bejövő minták címezték. A LUT-ok rekeszei az előtorzított mintákat tartalmazták. Ezeket a LUTokat a processzor forráskódjában tárolt együtthatókkal lehet inicializálni tetszőleges tartalmúra, majd a ZedBoard kapcsolóival váltani az előtorzított kimenet és a sima, „drótszerű” átvitel között.

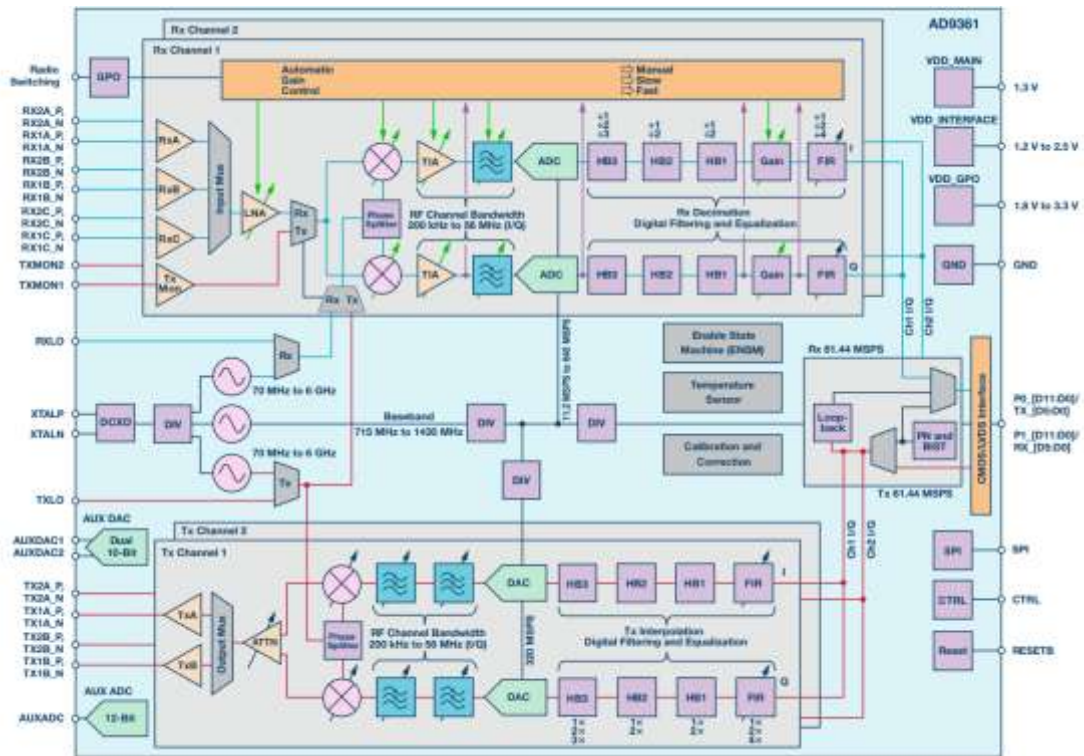
A következőekben részletesen bemutatom a megvalósított hardver és szoftver komponenseket. Kitérek az RF chip működésére, részegységeire. Bemutatom, hogy milyen szoftveres kalibrációkat kell lefuttatni a chipen ahhoz, hogy működőképes legyen.

A használt FPGA tulajdonságait, korlátait is megadom, illetve ismertetek néhány FPGA-s tervezési módszert, melyeket használtam a diplomatervezés során.

8.1 Az AD9361-es RF Chip tulajdonságai

Forrás: [15][15], [16]

Az AD9361 egy nagy integráltságú két csatornás adóvevő chip. 70MHz-től 6GHz-ig használható akár 56MHz-es sávszélesség mellett (nem minden működési konfigurációban). 12-bit-es AD-DA konverterekkel rendelkezik. Az alapsávi mintavételezett IQ jelet a CMOS/LVDS digitális interfácen keresztül tudja továbbítani az FPGA-nak. Tekintsük át működését a következő blokk diagramon:



8.2. ábra: AD9361 blokk diagram.

Az FMCOMMS-3 panel az Rx1A, Tx1A és Rx2A, Tx2A be- és kimeneteket használja. A két csatorna egyforma hardveresen. A vevő oldalon a bejövő jel egy AGC-vel ellátott LNA-n halad keresztül (AGC szoftveresen tiltható, konfigurálható). A következő lépésben az egyszeres keverésű vevőkre jellemző módon az RF jelet kétfelé bontva egymáshoz 90° fáziskülönbségű lokál jellel szorozza meg az IC. Így állnak elő a jel I és Q mintái melyek ettől kezdve külön kerülnek feldolgozásra, továbbításra. Az állítható törésponti frekvenciájú, egy pólusú TIA szűrőn áthaladva az alapsávi jel mintavételezésre kerül. Az, hogy milyen mintavételi frekvenciával történik ez meg, szoftveresen állítható és sok dolog befolyásolja. Az egyik ilyen a vételi decimáló szűrő blokk konfigurációja a másik pedig a digitális interface sebessége.

A lokáljelet több féle forrásból is elő tudja állítani az IC. Az egyik lehetőség, hogy RXLO és TXLO lábakra saját, külső forrásból adunk egy megfelelő jelet. Ennél kényelmesebb a XTALP és XTALN lábak közé kötött kvarc kristály (tipikusan 40MHz) rezgéséből chippen belül előállítani a lokál jelet.

Miután a jel mintái áthaladtak a vételi oldal FIR szűrőin (szoftveresen áthidalhatóak), a digitális interface továbbítja őket a CMOS/LVDS vonalakon. Mivel a

digitális interfésznek is számos beállítása van ezért képes különböző hardver kialakításokhoz illeszkedni. Különböző feszültség szinteken képes meghajtani az adat vonalakat, hogy illeszkedjen az FPGA feszültség szintjeihez. Képes differenciális és asszimmetrikus módban kommunikálni az FPGA-val. Differenciális módban kétszeres bitsebességet kell használni mivel fele annyi vonal áll rendelkezésre. A differenciális átvitel hosszabb jelutakat tesz lehetővé a PCB-n. A digitális átvitelt az RF csatornák konfigurációjától is függ.

Az adó oldalon rendkívül hasonló jelúttal találkozunk, mint vételi oldalon. Itt is van egy szoftveresen konfigurálható FIR és további interpoláló szűrők. A DAC-ok után szoftveresen hangolható aluláteresztő szűrők következnek és végül az I/Q keverő. A vételi és adási frekvencia különbözhet egymástól, de a két csatorna frekvenciája sajnos nem. A kimeneti erősítő erősítése, mint majdnem minden más is, szoftveresen állítható működés közben.

Az IC az egyenes keverésű adóvevők közé tartozik. Szemben a szuperheterodin vevőkkel, a homodin vevők nem használnak közbülső (IF) frekvenciát a hasznos jel kondicionálására és szűrésére. Az egyszeres keverésű vevők kevesebb alkatrészt tartalmaznak, és erősen támaszkodnak szoftver rádiós, digitális jelfeldolgozási módszerekre. Ezek a módszerek flexibilis megoldásokat és egységes hardver architektúrákat eredményeznek, melyek manapság egyre fontosabbá válnak.

Tekintsük röviden át az egyszeres keverésű vevők elméletét, tulajdonságait.

A homodin adóvevők a rádiójel vivőfrekvenciájával megegyező lokál jellel szorozzák meg az alapsávi jelet és egy lépésben keverik fel/le azt.

Előnyök:

- Alacsonyabb alkatrész szám, mely alacsonyabb áramfelvételt és költséget jelent.
- Kevesebb, olcsóbb szűrőre van szükség

Hátrányok:

- Az analóg tartományban pontatlan I/Q fázis és erősítés viszonyok esetén az oldalsáv elnyomás drasztikusan romlik
- Lokáljel szivárgás fordulhat elő

Hogy jobban érthető legyen az I/Q komponensek hibája tekintsük egy általános, digitálisan moduláció két 90° -kal eltolt vivőjét melyek A és B amplitúdójuk θ fázishibával.

$$x(t)_{IQ} = A\cos(\omega_0 t + \theta) + jB\sin(\omega_0 t) \quad (8.1)$$

Euler alakba átírva az egyenlet jobb oldalán szereplő kifejezéseket azt kapjuk, hogy:

$$A\cos(\omega_0 t + \theta) = \frac{1}{2}(Ae^{j\omega_0 t} e^{j\theta} + Ae^{-j\omega_0 t} e^{-j\theta}) \quad (8.2)$$

$$B\sin(\omega_0 t) = \frac{1}{2j}(Be^{j\omega_0 t} - Be^{-j\omega_0 t}) \quad (8.3)$$

Visszahelyettesítve:

$$x(t)_{IQ} = \frac{1}{2}(Ae^{j\omega_0 t} e^{j\theta} + Ae^{-j\omega_0 t} e^{-j\theta} + Be^{j\omega_0 t} - Be^{-j\omega_0 t}) \quad (8.4)$$

Feltéve, hogy sikerül fázis és amplitúdó helyesen elvégezni a modulációt és a demodulációt (tehát $A=B$ és $\theta = 0$ val):

$$x(t)_{IQ} = \frac{1}{2}(Ae^{j\omega_0 t} + Ae^{-j\omega_0 t} + Ae^{j\omega_0 t} - Ae^{-j\omega_0 t}) \quad (8.5)$$

Ami nem más, mint:

$$x(t)_{IQ} = \frac{1}{2}(Ae^{j\omega_0 t} + Ae^{j\omega_0 t}) = Ae^{j\omega_0 t} \quad (8.6)$$

Ebben az esetben tökéletes a tükör-elnyomás. Csak a pozitív frekvenciához tartozó komponensek lesznek jelen a spektrumban. Ahhoz, hogy ilyen lehessen az RF chip jele szoftveres kalibrációkat kell lefuttatni minden bekapcsoláskor.

A lokáljel szivárgása alatt azt a jelenséget értjük, amikor a lokál szintézer jele nem csak a keverő erre dedikált portjába jut el, hanem a RF jelébe is. Alapsávon egy

jelentős DC szintként fog megnyilvánulni mely szaturációba viheti az alapsávi erősítőláncot, illetve kisugározva szennyezi a spektrumot. Ezt is szoftveresen kell kikalibrálni az inicializáció során.

Az előbbi elméleti kitekintő után térjünk vissza a AD9361 chip működéséhez. Az összes beállítást, kalibrációt és parancsot lehetséges SPI-on keresztül elküldeni az áramkörnek. Ezek az utasítások mozgatják különböző állapotok közt a chip állapot gépét. Ez az állapot gép adja meg, hogy milyen módban kommunikál a chip, egyáltalán kommunikál-e, vagy kalibráció alatt van. Természetesen van reset lába és egyéb külső vezérlő vonalai. Ezek a vonalak valósídejű üzemmód váltásokat tudnak kiváltani. Programozható bit kimenetekkel is rendelkezik a chip, melyek speciális alkalmazásokban az FPGA számára adhatnak az SPI-nál gyorsabb jelzéseket.

A következő kalibrációkat kell az adóvevő használatához elvégezni induláskor:

- BBPLL VCO kalibráció
- RF szintézer töltéspumpájának kalibrációja
- RF szintézer VCO-jának kalibrációja
- Alapsávi Rx analóg szűrők kalibrációja
- Alapsávi Tx analóg szűrők kalibrációja
- Alapsávi Tx másodlagos szűrők kalibrációja
- Alapsávi DC offset kalibrációja
- RF DC offset kalibrációja
- Rx és Tx kvadratúra kalibrációja

Látható, hogy az áramkör meghaladja azt a komplexitást, amit egy fejlesztő még képes kezelni egyedül, betartva az iparban szokványos határidőket. Mivel a fejlesztőkártya célja és értelme, hogy gyorsan lehessen prototípusokat előállítani használatával ezért a gyártó készített szoftver és hardver csomagokat melyekkel sok terhet levesz a felhasználó válláról. Használatukkal nem szükséges az eszköz minden apró részletét ismerni és néhány nap után már bele is lehet fogni a tényleges fejlesztésbe.

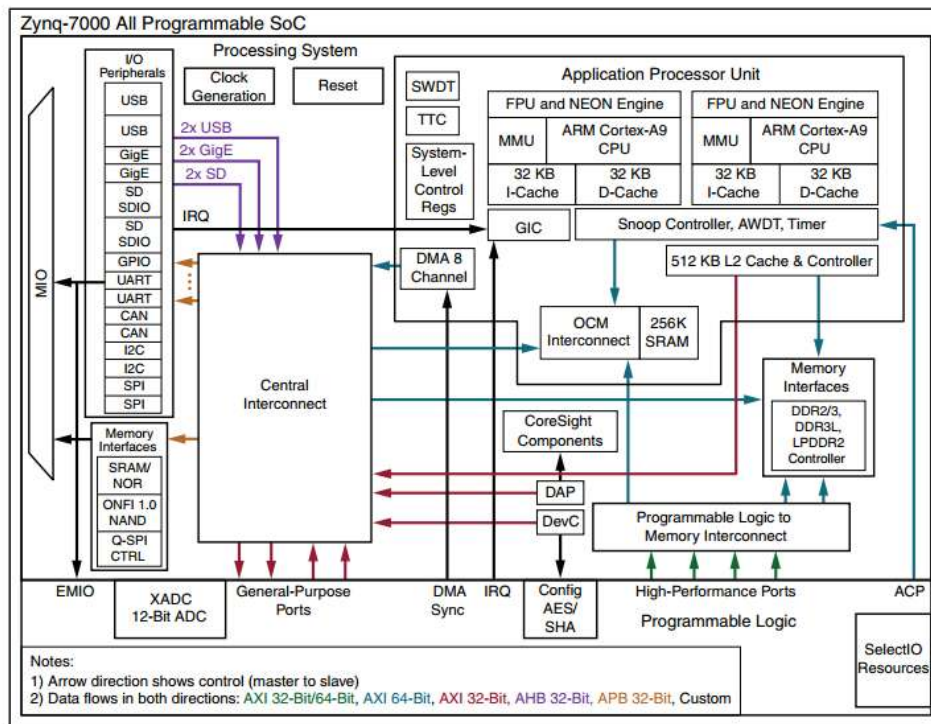
8.2 A Zynq Z-7020-as SoC tulajdonságai és a fejlesztés menete

Forrás:[17][18]

A Z-7020-as áramkör egy sokoldalú, konfigurálható beágyazott számítási architektúra.

Két részből áll: Programmable System (PS) és Programmable Logic (PL).

Kevés olyan DSP funkció van, amit ne lehetne implementálni ezen az áramkörön. A PS perifériakészlete minden, beágyazott környezetben használatos, kommunikációs módot lefed. A PL számítási kapacitása pedig igen komplex feladatok gyors végrehajtására is alkalmassá teszi a SoC-ot. A lenti ábrán látható az ARM magok és a PL kapcsolata illetve a SoC hard core periféria készlete:



8.3. ábra: Zynq-7000 SoC.

Az általam készített jelfeldolgozó áramkör erősen épít a PL Block RAM-jaira. A BRAM egy olyan dedikált memória mely rugalmasan használható egyedi hardverek tervezésekor. Ellentétben a PL flip-flopjaival, nem használható más funkció megvalósítására, mint memória. Az előtorzítóban „Simple Dual-Port Block RAM”-okat használtam, mert egyik porton mindig írtam az Rx1,Rx2 jelek mintáit a másik porton mindig olvastam azokat. A dual portos BRAM-ok kiválóan alkalmasak különböző

órajel tartományok összekapcsolására. A BRAM-ok példányosításának legegyszerűbb módja, ha Vivado „Language Template”-ei közül kiválasztjuk a megfelelő példakódot, és létrehozunk neki egy saját modult.

A SoC részegységei AMBA AXI buszon kommunikálnak egymással. Mivel az AXI busz a preferált módszer minden periféria integrálása esetén ezért én is ilyen módon kapcsolom hozzá a jelfeldolgozó célhardveremet az APU-hoz. Ehhez természetesen szükség volt megfelelő fejlesztőkörnyezetekre. A hardver fejlesztését Vivado 2015.2-ben, a szoftverekét pedig Software Development Kit (SDK) 2015.2-ben végeztem.

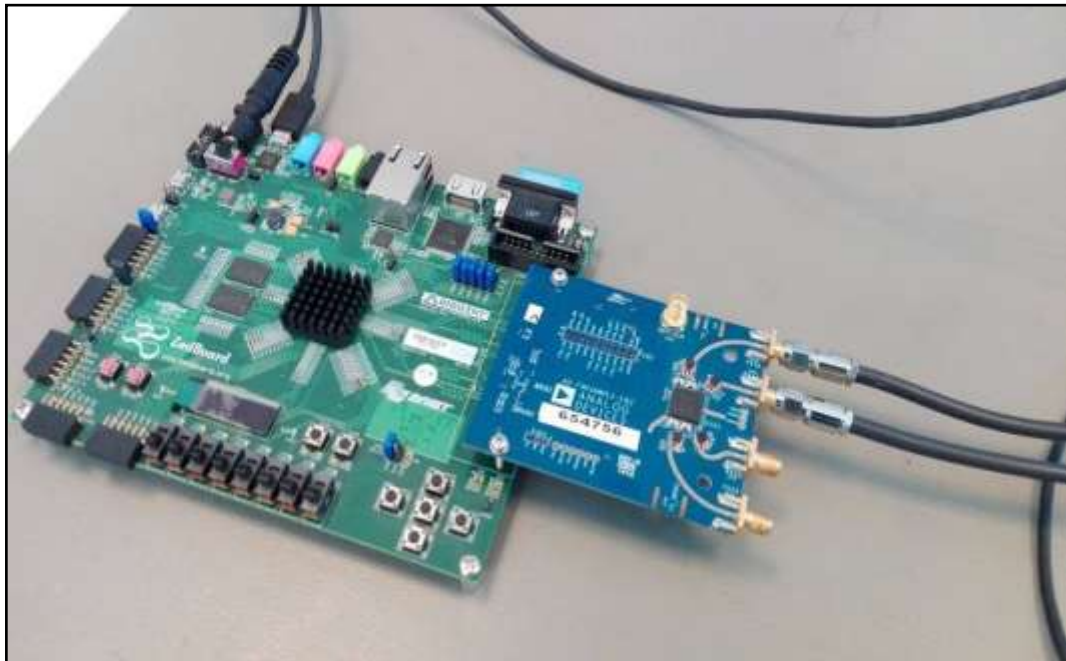
A Vivado GUI alapú hardvertervezés a következő lépésekből áll:

- I. Létre kell hozni egy projektet a használt fejlesztőpanelhez tartozó előre definiált beállításokkal. Ilyen beállítások az FPGA pontos típusa, tokozása, lábkiosztása.
- II. Létre kell hozni egy Block Design-t a projekthez és intuitív módon a különböző használni kívánt IP-eket hozzáadni. Ha egyedi perifériákat akarunk használni, akkor érdemes azokat egy általunk létrehozott IP könyvtárban elkészíteni, kompatibilis AXI-busz interfacekkel. A processzoros rendszer is néhány IP hozzáadásával elkészíthető.
- III. Verifikáció szimulációval. Mivel az elkészített IP-eket elégséges sokszor önmagukban verifikálni ezért a buszrendszer szimulációját kihagytam.
- IV. Ha megadtuk az összes szükséges fizikai „megkötést”, a szükséges IP-k HDL file-jai (Verilog vagy VHDL) hibátlanok és a szintézis beállításai is igényeinknek megfelelőek akkor szintézist elindítható. A szintézis során a program előállítja azokat a Net listákat, amik leírják a logikai kapcsolatokat az egyes elemi hardverkomponensek között. Az absztrakt HDL file-okból a szintézis készít implementálható rendszer leírást.
- V. Az implementáció során a program megpróbálja hozzárendelni a Net lista elemeit tényleges erőforrásokhoz az FPGA-ban (mapping). Ha sikerült ez a lépés, akkor össze kell kötni az egyes elemeket egymással (routing). Ez egy nagyon időigényes folyamat és érdemes minél kevesebbszer elvégezni. Az implementáció végére kiderül, hogy sikerült-e az időzítési feltételeket teljesítenie a hardvernek. Ha nem, akkor további módosítások szükségesek. Ezekről a lehetséges módosításokra később írok.

- VI. Bitfile generálás során létrejön az a néhány MB méretű állomány mellyel felkonfigurálható az FPGA.
- VII. Ha elkészült a hardver, akkor kezdődhet a szoftverfejlesztés. Exportálni kell a projektet SDK-ba, ami készít a fejlesztő számára egy Board Support Package-et. Ezek a file-ok tartalmazzák a processzoros rendszer inicializációs kódját, a periféria címeket és bizonyos perifériák drivereit. A saját egyedi perifériáink drivereit és végrehajtandó program kódot magunknak kell megírni természetesen. Külső forrásból származó .c és .h file-okat is importálhatunk a projektbe.

8.3 Az FMCOMMS-3 panel illesztése ZedBoardhoz

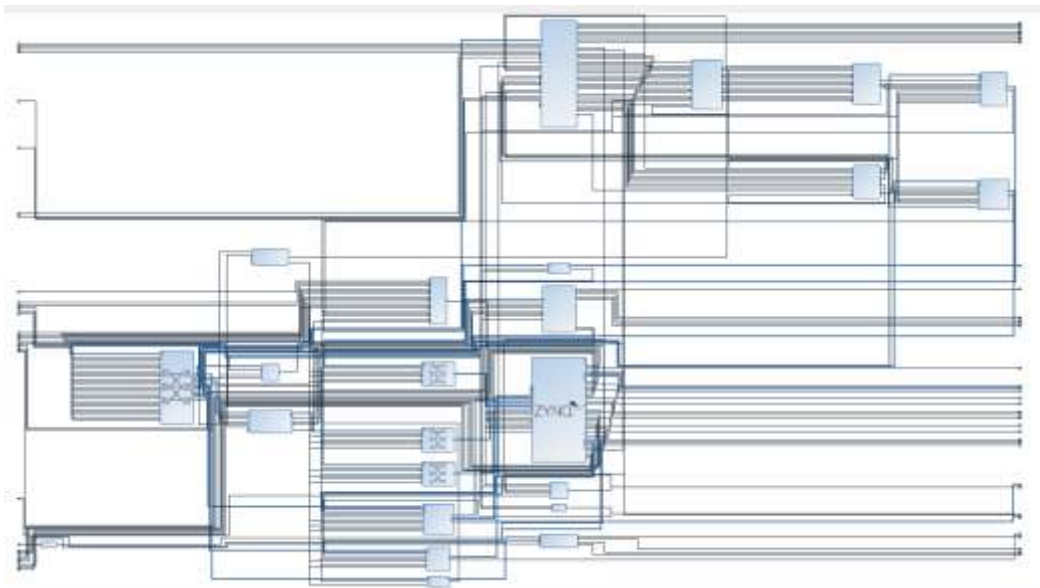
A gyakorlati megvalósítás első lépéseként össze kellett állítanom a 8.1. ábrán látható keretrendszer magját. Mivel a két fejlesztőpanelt interoperábilisra tervezték ezért nem csak fizikailag lehetséges összekapcsolni őket, hanem az Analog Devices folyamatosan fejleszt referencia dizájnokat a termékük használatához.



8.4. ábra: A ZedBoard és a hozzákapcsolt FMCOMMS3.

Első próbálkozásaim során ezeket a referencia dizájnokat teljesen figyelmen kívül hagytam és az adatlapok alapján újra szerettem volna írni minden szoftver és hardver komponenset. Több elpazarolt hét után beláttam, hogy ez egyszerűen nem kivitelezhető. Minden szempontból túl bonyolult és alul dokumentált az AD9361-es

chip az ilyen jellegű felhasználáshoz. A javasolt tervezési metódus szerint le kell tölteni az általunk használt Vivado verzióhoz készített könyvtárakat, melyek olyan IP-eket tartalmaznak, amikből összeállítható a referencia hardver. Ezt build-elésnek hívja a gyártó. Itt is számos problémába futottam. Kiderült, hogy a Xilinx olyan módosításokat végez a különböző verziójú Vivadokon, hogy ugyan az a TCL script, ami hibátlanul fut egy adott verzión, nem képes lefutni a frissebb verziókon. A referencia project dokumentációja alapján sikerült egy példa projektet összeállítanom. A project blokk dizájnja a következő ábrán látható:

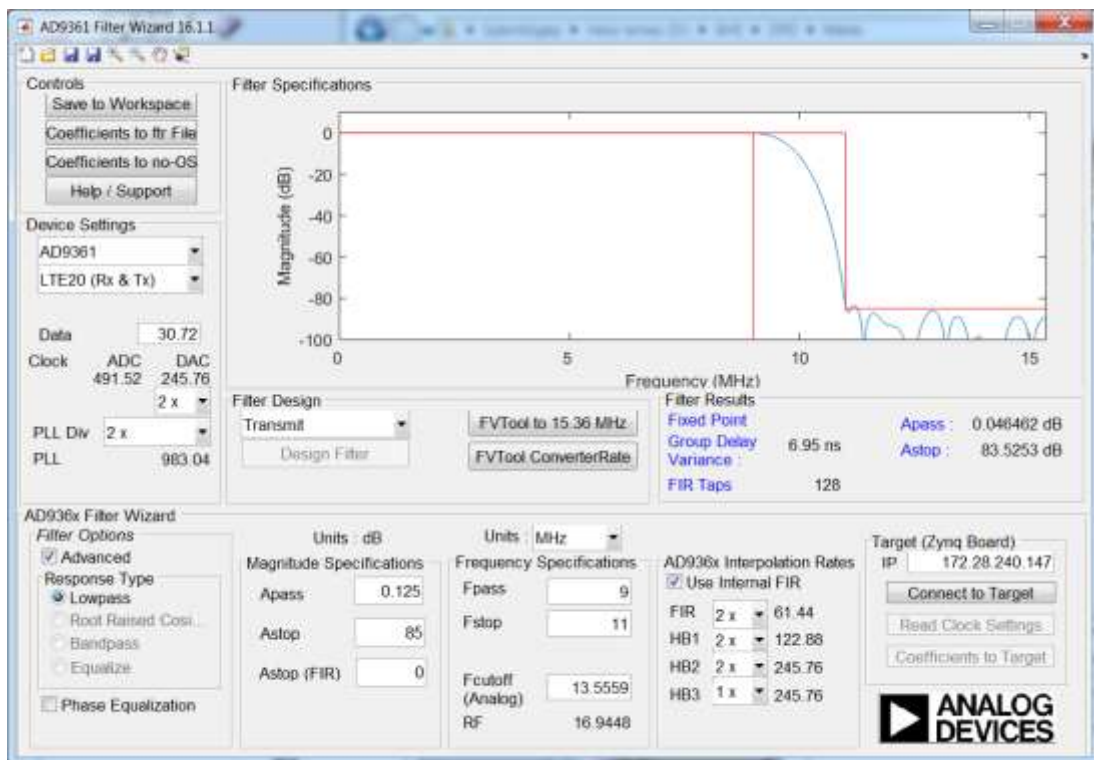


8.5. ábra: A példa projekt

Az Analog Devices alapgondolata az volt, hogy egy olyan rendszert terveznek, ami mindenre jó. Képes Linux operációs rendszert futtatni, HDMI bemenetű monitort meghajtani és a jel spektrumát rajta valós időben megjeleníteni. Képes egy szinuszt előállítani és a kimenetre adni, a jel mintáit a PS memória terében mozgatni DMA átvitelrel. További haszontalan funkció, hogy az alapsávi adatokat, mint audio jelet is lehetséges meghallgatni. A számomra legfontosabb funkciót, hogy az FPGA-ban elérhetővé teszi a bejövő jel mintáit és visszaküldi a kimenetre azonban csak hardveres és szoftveres módosítások után tudta teljesíteni. Ez a rendszer határozottan marketing célokat szolgál, és az igazi fejlesztők igényeit csak sok munka és változtatás után képes kielégíteni. A projekt dokumentációja az olvasóra bízta a különböző részegységek, IP-k funkciójának értelmezését ezért az sem egyértelmű, hogy pontosan mi mire való. A termék supportjával történő sorozatos egyeztetések és számtalan fórumkutatással töltött

nap után sikerült kikapcsolnom a belső DDS-t, a PS memória terébe történő adatmozgatásokat és felülírni egyéb gyári beállításokat.

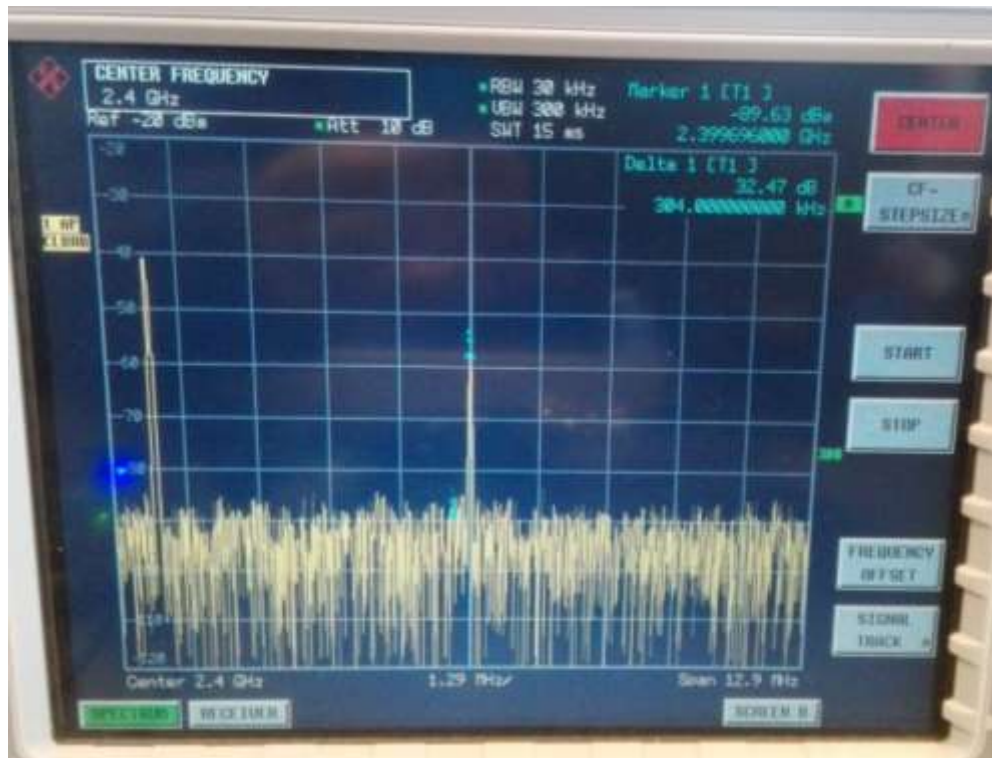
Ha a példa projekt hardveresen nehézkesnek és túlbonyolítottnak tűnik, akkor a szoftver komponensek kifejezetten átláthatatlanok, bárminemű módosításuk igen türelmes mérnököt kíván. Akárhogy próbáltam nem tudtam az RF chip FIR szűrőit felkonfigurálni saját együtthatókkal, pedig még a gyártó Matlab App-ját is letöltöttem hozzá. Egy valós alkalmazásban a FIR szűrők hiánya meg nem engedhető spektrumszennyezést is okozhatott volna. Természetesen azok a szűrők helyettesíthetőek az FPGA erőforrásainak felhasználásával, de a példa projekt már önmagában is sokat felhasznál belőlük. A legbosszantóbb, hogy nem moduláris a project, eltávolítva fölösleges komponenseket az esetek többségében működésképtelenné válik. A PL DSP blokkjainak körülbelül 22% így egy olyan DDS implementációjára megy el, amit soha nem fogok használni.



8.6. ábra: A FIR szűrő tervező Matlab App.

Nem kétlem, hogy az App működőképes és az is biztos, hogy helyes C kódot generál, de a szoftver bonyolultsága illetve dokumentációjának teljes hiánya komolyan gátolja a használhatóságát.

A FIR szűrők kikapcsolása nem okozott jelentős problémát a munka során. A több ezer sornyi kalibrációs kód majdnem tökéletesen működött. Azonban volt egy probléma, amit nem tudtam kiküszöbölni. A Tx oldal vivőfrekvenciája nagyon magas szinten jelen volt a kimeneti spektrumban. Ez vagy a lokáljel szivárgásra vagy a DC offset kalibráció nem kielégítő mértékére utal.



8.7. ábra: A Tx lokál jel szivárgása (marker 1) és egy szinusz.

Ezen a ponton sikerült egy olyan FPGA alapú rendszert összeállítanom, ami képes (ha nem is kiemelkedő jeltisztaság mellett) nagyjából 30MHz széles sávban átvinni rádió jeleket úgy, hogy a jel mintáinak alapsávi feldolgozása, módosítása lehetséges valós időben.

8.4 Egyedi perifériák készítése

Az előtorzító realizálásának számos lehetséges módja van. A legkézenfekvőbb ezek közül a megoldások közül egy olyan lett volna, ami a PL DSP blokkjaira épül. Ezek szorzó áramkörök melyek képesek két 18 bites operandusból egyetlen órajel alatt előállítani a 36 bites eredményt. Mivel a szimulációk alapján ötöd, illetve akár heted fokú polinom is szükséges lehetett volna az erősítő modellezéséhez, ezért számábrázolási gondokba ütköztem volna azonnal ezzel a módszerrel. Az effektív

bitszám gyors csökkenése, a szaturálás problémája, abszolút-érték képzés, illetve előjel kiterjesztés szükségessége miatt arra jutottam, hogy más megközelítést alkalmazok. A bejövő jel mintáinak hatványozása és konstanssal való szorzása sokszor csak a részeredményekben vezet irreálisan sok biten ábrázolható számokra. Ezért az összes kettes komplementes módban ábrázolt 12 bites szám előtorzított mintáját meghatároztam az a_{kl} vektor megfelelő együtthatójának felhasználásával. Mivel mindezt lebegőpontos aritmetika használatával tettem Matlabban, offline ezért sok számábrázolási problémát kikerültem. Így K darab LUT keletkezett, melyek megfelelő elemeit minden új minta érkezésekor össze kell adni, de nem kell szorzásokat végrehajtani. A K darab LUT-ot a bejövő minták címzik.

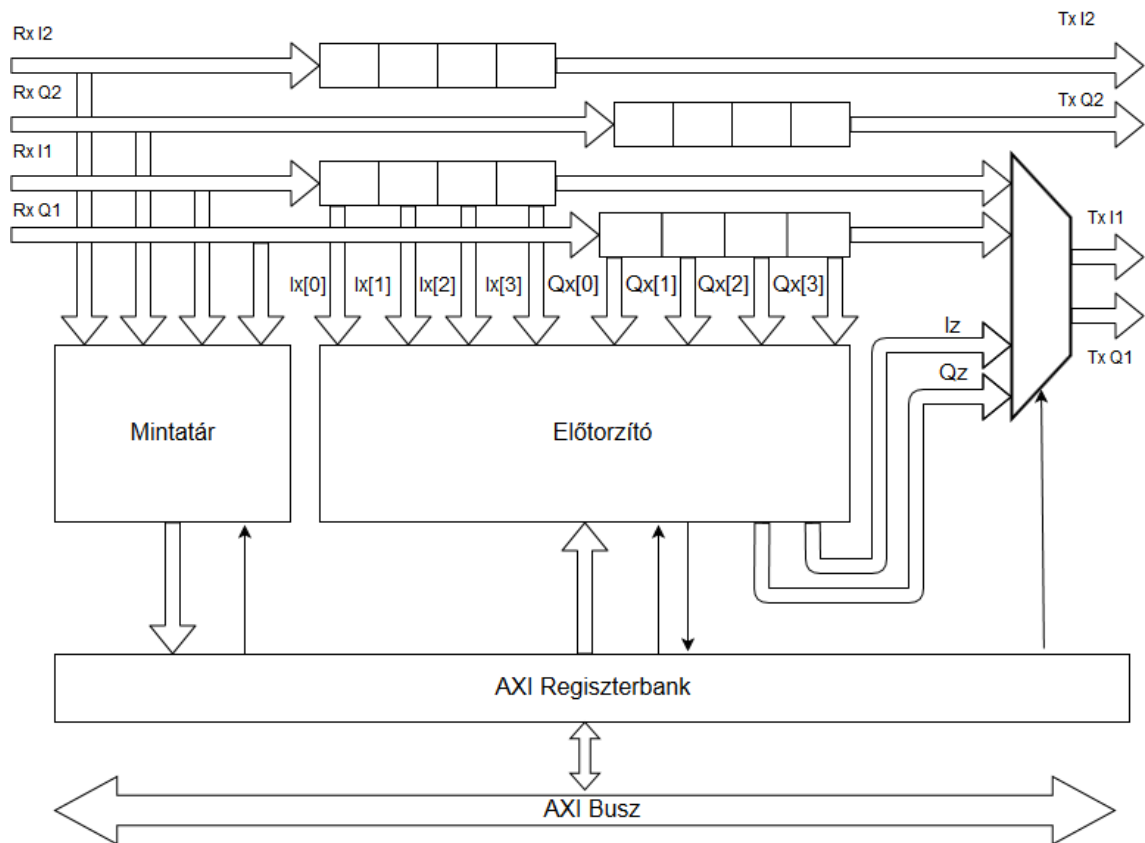
$$w_k = \sum_{l=0}^{L-1} a_{kl} \Phi_{kl} = \sum_{l=0}^{L-1} a_{kl} x_{[n-k]} |x_{[n-k]}|^l \quad (8.7)$$

Mivel hardveresen csak K darab szám összeadását kellett kezelnem sokkal könnyebb helyzetben voltam. A gyakorlatban K=4 választással éltem, ezt a számot a szimulációs eredmények- és az elérhető memóriára alapján választottam.

$$z_{[n]} = \sum_{k=0}^3 w_k \quad (8.8)$$

Tehát a 8.7-et kiértékelem k=0...3-ig minden x=-2048...+2047 értékre, és egy LUT-ba mentem melyet a PS tölt be a saját forráskódjából a BRAM-okba. Érdeemes megjegyezni, hogy ez nem jelenti automatikusan, hogy csakis négy minta memóriával rendelkező modellekre szorítkoztam. Amennyiben csak a k=0-hoz tartozó LUT-ot töltöm fel értékes számokkal és a többi csupa 0-val, akkor egy memória nélküli rendszert modellezek.

A fentiekben részletezett digitális áramkör blokkvázlata a 8.8.-as ábrán látható.



8.8. ábra: A digitális előtorzító hardver blokkvázlata.

Tehát adott az előtorzítás módja, azonban ez még nem a projekt legnehezebb része. A valódi probléma a megfelelő a_{kl} vektor megtalálása. Az ILA architektúra épít a bejövő és a visszacsatolt jel ismeretére is. Ezeknek a jeleknek az I és Q komponenseiből egy 8196 hosszú részt letárolok BRAM-okba. A BRAM-ok az AXI Lite interfácen keresztül olvashatóak. Ahhoz, hogy Matlabban elő tudjam állítani az a_{kl} vektort (majd az előtorzított mintákat a LUT-ok számára), szükséges a PS memóriájából a mintákat UART-on a TerraTerm programmal szöveges fájlba menteni. Természetesen lehetséges lenne a kommunikációt interaktívabb módon megvalósítani az FPGA és a Matlab között, de ez módszer volt a leggyorsabban működésre bírható. Miután elkészült a .txt fájl, azt egy Matlab scripttel megnyitom, beolvasom és feldolgozom. Az I és Q mintákból külön-külön függetlenül alkotok egy-egy memória polinom alapú erősítő modellt. A modelleket felhasználva összeállítom a LUT-okat és automatikusan C kódot generálok melyet elmentek egy txt fájlba. Az új txt fájl tartalmát beillesztve az SDK-ban található forráskódba (egy gyors fordítás után) újra programozom a Soc-ot, és a ZedBoard kapcsolóival betöltöm az új LUT-okat. A ZedBoard kapcsolóival lehetséges a

minták letárolását és a LUT-ok felkonfigurálását is kezdeményezni. Az előtorzítás szoftveresen (szintén egy kapcsolóval) bekapcsolható ilyenkor. A kapcsolókat és a hozzájuk tartozó LED-eket egy egyszerű, de egyedi GPIO periféria kezeli melyet szintén én írtam.

Amikor az előtorzító hardverét szintetizáltam Vivadoban, több időzítési problémára derült fény. Egy FPGA-ban az időzítési probléma azt jelenti, hogy az egyes részegységek nem kapják meg adott órajel élváltásra az új adatot ezért még a régi, már érvénytelenné vált, bitet (biteket) továbbítják, dolgozzák fel. Sokszor csak néhány jelút késik egy több bitet tartalmazó buszból. Ha ilyen adatokat továbbít, vagy ilyen bitvektorok alapján hoz döntést a hardver, akkor a hibás működés elkerülhetetlen. Minden bonyolult digitális áramkörnél felmerülnek időzítési gondok, ha ki akarjuk használni az alkalmazható (majdnem) maximális órajel frekvenciát. A problémának több lehetséges megoldása van:

1. Az időzítéssel kapcsolatos megkötések lazítása

Lehet, hogy a megírt hardver valójában alkalmas a hibátlan működésre, de túl szigorú időzítéseket írtunk elő, amiket nem fog tudni a fejlesztő környezet biztosítani pedig nincs is rájuk szükség. További pozitív hozadéka a módszernek, hogy a fejlesztőkörnyezet képes hatékonyabban erőforrásokat allokálni az egyes funkciókhoz, mivel nagyobb mozgásteret kap. Ilyenkor a rendszer, a hibaüzenetek ellenére, működni fog.

2. Az órajel frekvenciájának csökkentése

Ha kisebb órajelről jár az áramkör, több ideje van az adatoknak eljutni annak minden pontjára. Az első módszerrel együtt kell alkalmazni. A rendszer számítási teljesítményét csökkenti.

3. A dizájn módosítása, pipeline-osítás

Ha komoly időzítési probléma van, lehet rossz logika mentén terveztük meg a hardvert, és létezik racionálisabb megoldás is. Ha nincs akadálya érdemes regisztereket beiktatnunk az adat útba. Bizonyos pontokon egy órajellel megkésleltetjük az adatokat,

így a fejlesztő környezet nagyobb mozgásteret kap a jelterjedési idők optimalizálására. A számítási teljesítmény így nem csökken, csak a késleltetés nő.

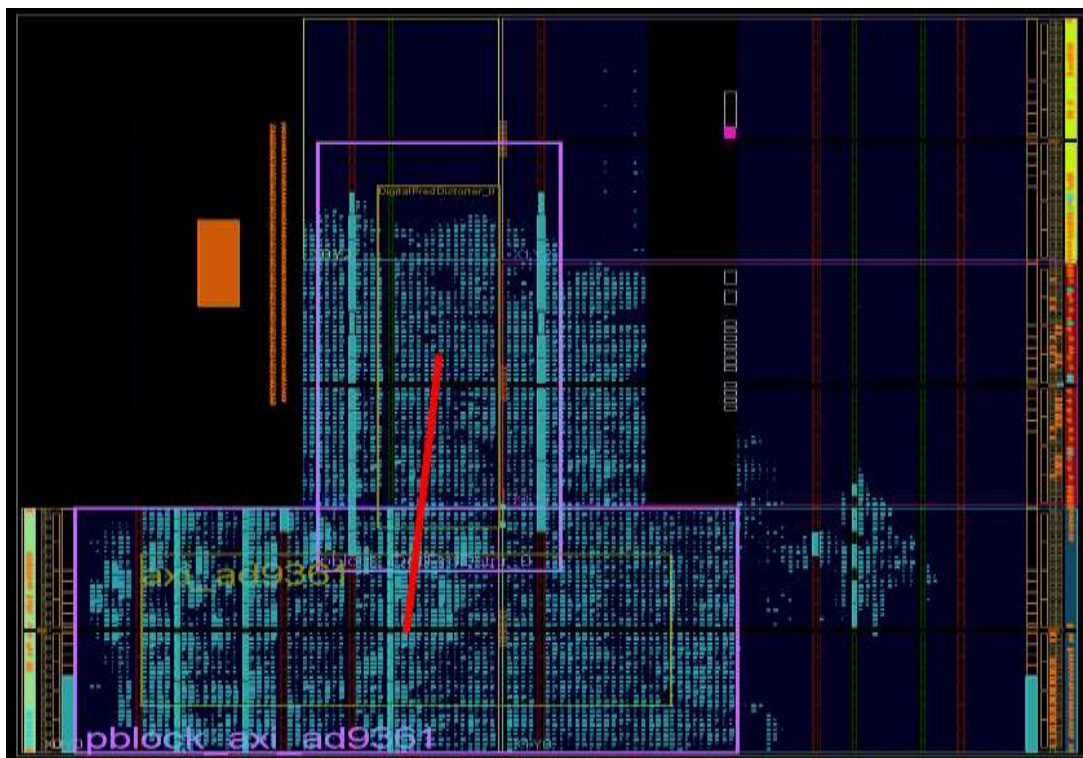
4. Floor planning

Az hogy implementációs fázisban hova kerülnek a különböző funkciójú modulok az FPGA-n belül sztochasztikus folyamat. A fejlesztőkörnyezetek nem tudják minden futásnál megtalálni az optimális elrendezést. Meg lehet adni constrain-eket a különböző IP-k fizikai elhelyezésére a chipen belül. Ezzel akár rosszabb designhoz is juthatunk, mint ha a programra hagytuk volna a feladatot, de sokszor egymáshoz közel tudjuk kényszeríteni az összetartozó IP-ket, ezzel csökkentve a jelterjedési időket.

5. Figyelmen kívül hagyás

Ez a legkockázatosabb és legkevésbé kifinomult problémakezelési stratégia. Ha a design olyan elemeinél jelez időzírási gondokat a környezet, amit nem használunk, vagy más okokból kifolyólag érdektelenek akkor dönthetünk úgy is, hogy nem változtatunk semmin.

Sok problémát okozott számomra is az adatok késése. Az RF chip referencia dizájnya nagyméretű és olyan constrain-t tartalmazott ami, 200MHz-es órajelhez tartozó időzítéseket ír elő az adatokat mozgó logikának. A valóságban csak 160MHz-es órajelet ad a chip az FPGA jelfeldolgozó részének, így ezen a megkötésen lazítottam. Mivel még így is volt néhány problémás regiszter a dizájnból ezért manuálisan átalakítottam a projekt floor plan-jét.



8.9. ábra: Manuális floor plan.

A 8.9.-es ábrán látható miként csoportosítottam két részre a logikát. A felső, állított téglalapban foglal helyet az előtorzító hardver. Mivel sok BRAM-ot használ ezért előnyös volt a két BRAM sávra helyezni, közel a narancssárga színnel jelölt ARM processzorhoz. A kép alján lévő fektetett téglalapban található az RF chip referencia dizájnja. A két blokkot úgy helyeztem el, hogy legyen metszetük. Láthatóan minimális a pblock-okon kívüli logika mennyisége.

Sajnos mindezek ellenére is volt néhány regiszter melyeknél az időzítési problémák nem oldódtak meg még így sem, de pipeline-osítással sikerült lecsökkentenem a jelterjedési időket ott is. Csak a referencia dizájn DDS-ében maradtak időzítési hibák melyeket nem tudtam kijavítani. A DDS –nek nincs haszna ebben az alkalmazásban így figyelmen kívül hagyhattam hibáit.

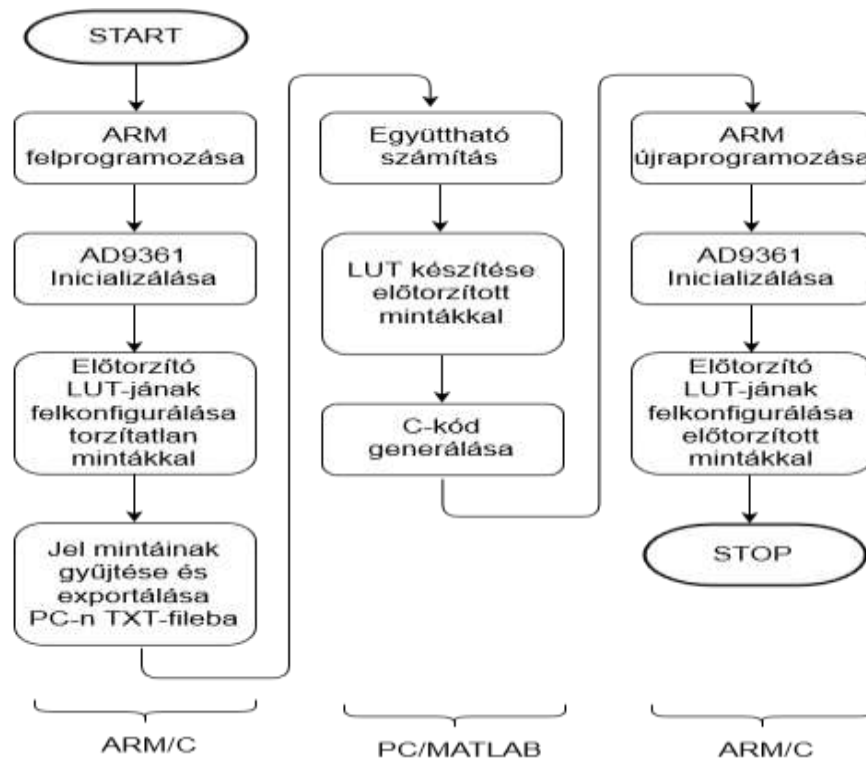
8.5 A beágyazott szoftver

A kész hardverek önmagukban működésképtelenek. A Zynq alapú tervezés lényege, hogy csak az időkritikus feladatokat implementáljuk hardveresen és minden egyéb funkciót szoftveresen valósítunk meg. Ennek a módszernek több előnye is van. A

hardveres hibák nehezen felderíthetők és megoldásuk nagyságrendekkel több időbe kerül, mint egy szoftveres hibáé. A szoftveres funkciók gyorsan átalakíthatóak, kibővíthetők.

A szoftver minden induláskor lefuttatja az AD9361 inicializálásához szükséges kódrészletet. Ez a kódrészlet SPI üzeneteket küld az RF chipnek és sorban elvégzi az összes kalibrációt. Az én módosításaimnak köszönhetően beállításra kerülnek az Rx ágak erősítései, a Tx ágak csillapításai és az ad9361_bist_loopback(...) függvény hatására a minták mozgatása az FPGA-n keresztül történik. Inicializáció után meghívom a main2() függvényt. Ez a függvény a kiindulópontja minden általam használt egyedi perifériavezérlő drivernek és alacsony szintű funkcionalitást megvalósító függvényének.

A következő folyamatábrán tekintsük át, hogy hogyan zajlik a rádiójel mintáinak gyűjtése, az együttható számítás és az előtorzított minták meghatározása.



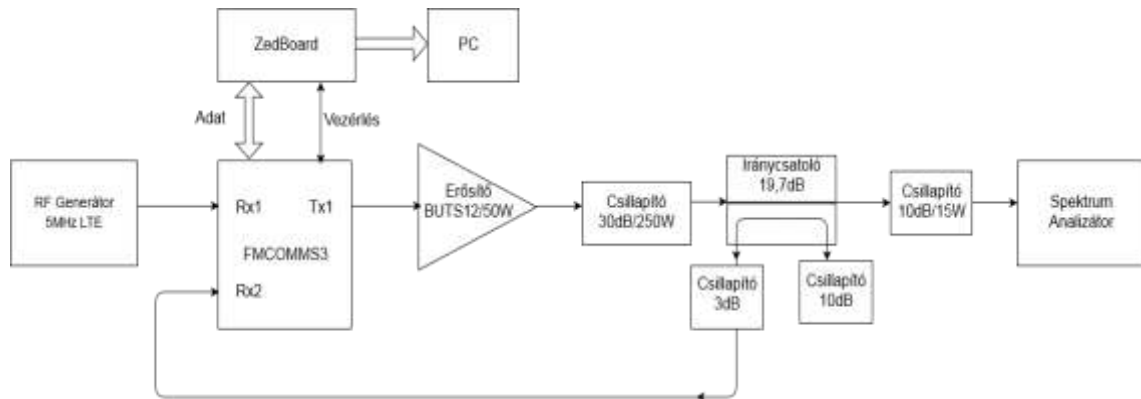
8.10. ábra: Az előtorzító LUT-jainak előállítása.

Az előtorzítás számításigényes részét offline végzem Matlabban. Az ARM szoftvere folyamatosan vizsgálja a kapcsolók értékét és a megfelelő felfutó él esetén elindítja a hardver Rx vonalainak tárolását. Amikor végzett a feladattal akkor a szoftver kiolvassa a mintákat és továbbítja UART-on őket a PC-re, ahol a 6.1.-es fejezetben leírtak szerint kiszámítom a szükséges együttható készletet. A bejövő és visszacsatolt

minták időkülönbségét keresztkorrelációjukból számolom ki (5.5.-ös fejezet). Az Rx2 (visszacsatolt) jel mintáiból képzett vektorokat ennek a különbségnek megfelelően (kb. 22 minta, de mérési elrendezéstől függ) elcsúsztatom a számítás pontosságának érdekében. Írtam egy Matlab scriptet, mely az együtttható készlet alapján C forráskódot generál. Ezzel a kódrészlettel kibővítve az ARM processzor kódját újra fordítom azt, megint felprogramozom a processzort és eltorzítom a kimenetet.

9. Mérések, előtorzítás

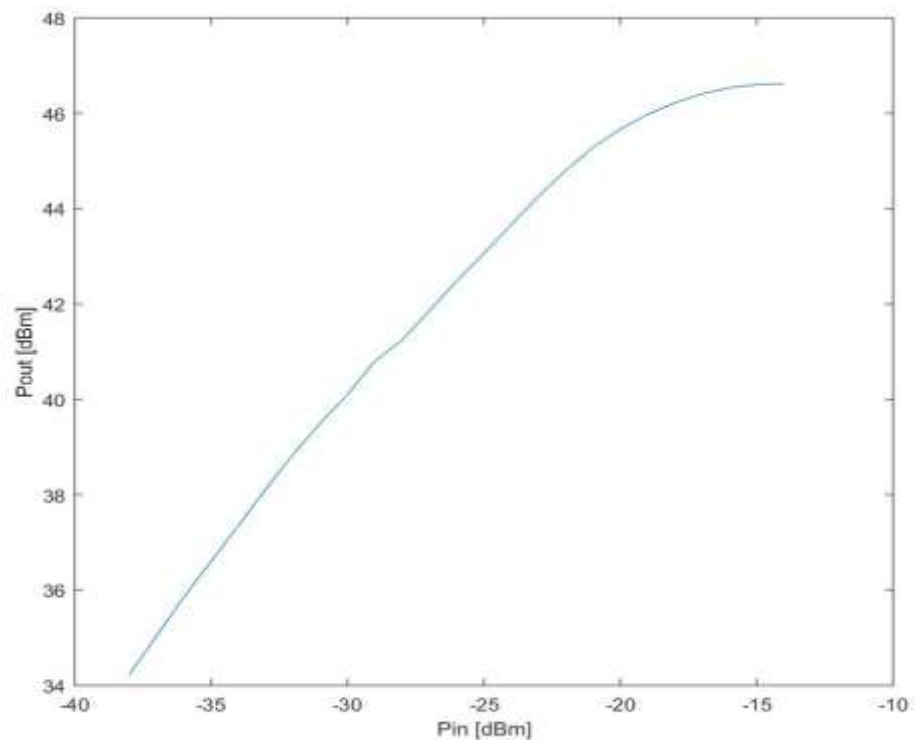
A következő mérési elrendezést használtam a jelfeldolgozó áramkör vizsgálata során:



9.1. ábra: A mérési elrendezés blokkvázlata.

A fenti ábrán látható rendszert nagy csúcstényezőjű LTE jellel gerjesztettem, melynek sávszélessége 5MHz, vivőfrekvenciája 2,2GHz. A mérési elrendezés két legfontosabb része az FMCOMMS3 panel és a Bonn Hungary Kft. 50W-os teljesítmény erősítője. Nagy körültekintéssel kellett megválasztani a mérő rendszer összes elemét, mert az erősítő kimenő teljesítménye képes tönkretenni bármelyik műszert vagy mérési eszközt.

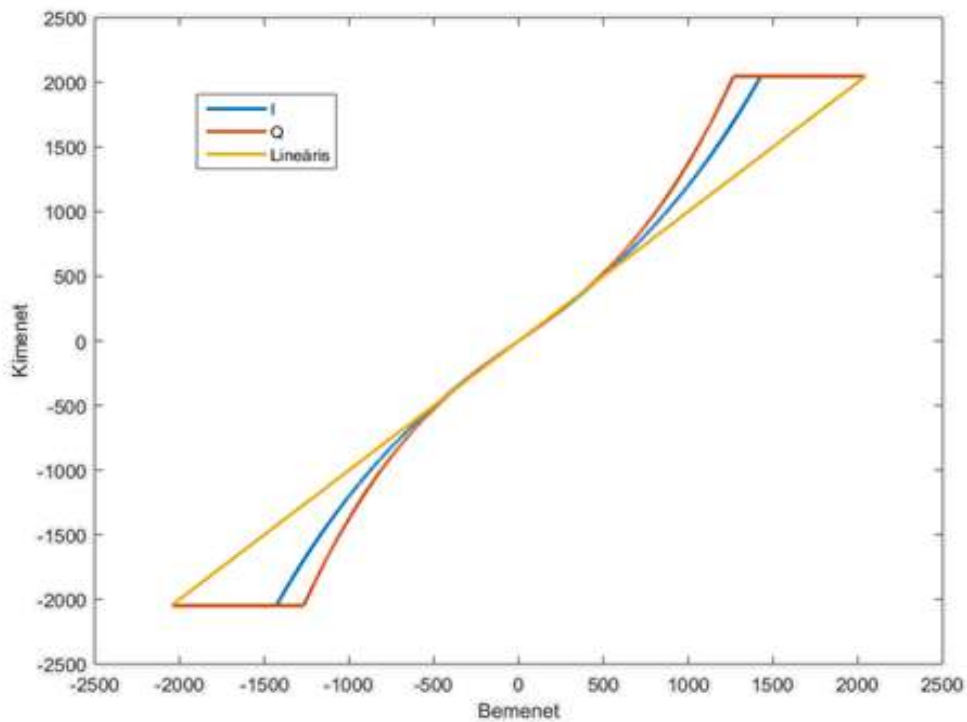
Az előtorzító vizsgálata előtt számos mérést végeztem, hogy az RF panel két bemenetén ugyan akkora jelszint legyen. A BUTS12 erősítő karakterisztikáját is felvettem egy másik (sokkal egyszerűbb) mérési elrendezésben, hogy tudjam, körülbelül mekkora jelszinttel lehet elhagyni karakterisztikájának lineáris szakaszát. Az erősítő 46dBm-nél kezd el szaturálni.



9.2. ábra: Az erősítő linearizáció előtti karakterisztikája.

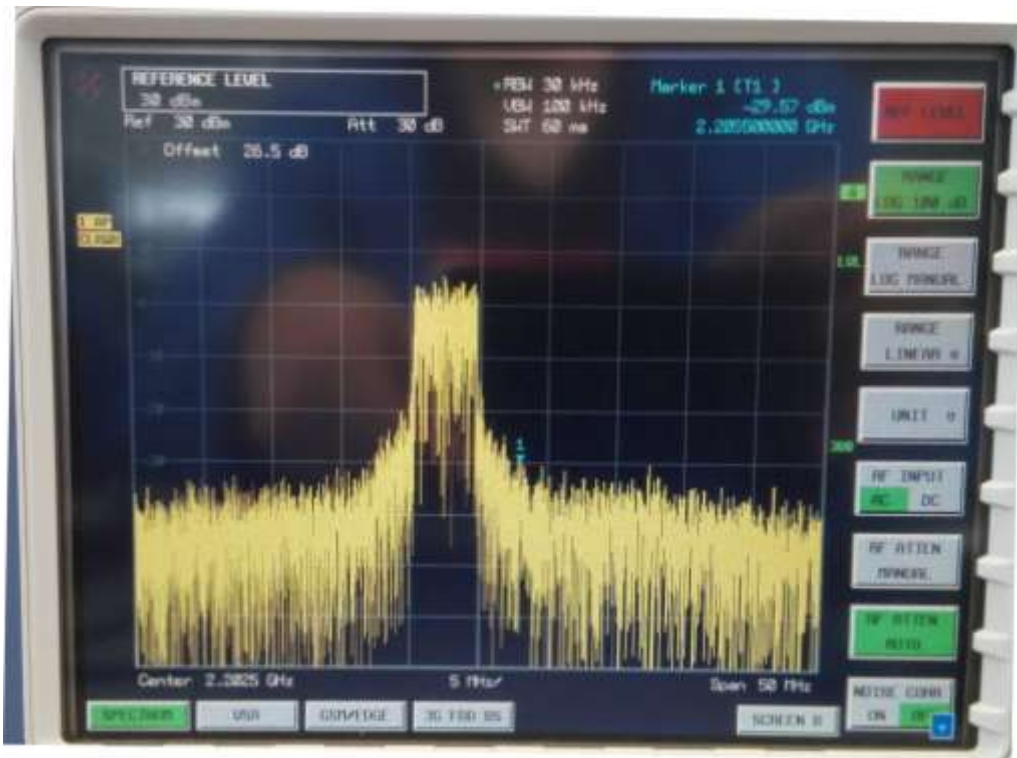
Sajnos minden próbálkozásom ellenére sem sikerült az erősítő karakterisztikáját lineárisabbá tennem. A jelfeldolgozó hardver és szoftver pontosan azt csinálja, amire terveztem, de a módszer elméleti részében olyan modellezési hibát véhettem mely következtében nem sikerült linearizálnom az erősítőt.

A 9.3. ábrán látható, hogy az előtorzító (a várakozásoknak megfelelően) nem lineárisan megváltoztatja az alapsávi jel I és Q mintáinak amplitúdóját, méghez hozzá különböző mértékben. Elméletben így nem csak az erősítő amplitúdó, hanem fázishibáját is korrigálni lehetne. Az előtorzító a 9.3. ábrán egy memóriamentes harmadrendű rendszer inverzét próbálja modellezni 5000 mérés alapján. A kimeneteket szaturálom, amikor meghaladják a DA konverterek számábrázolási tartományát. Ezeket a szaturált görbéket tárolom le az előtorzító LUT-jaiba. A bejövő minták megcímzik a LUT-ok memória rekeszeit (x tengely) és a kimenetre a LUT tartalma kerül (y tengely).

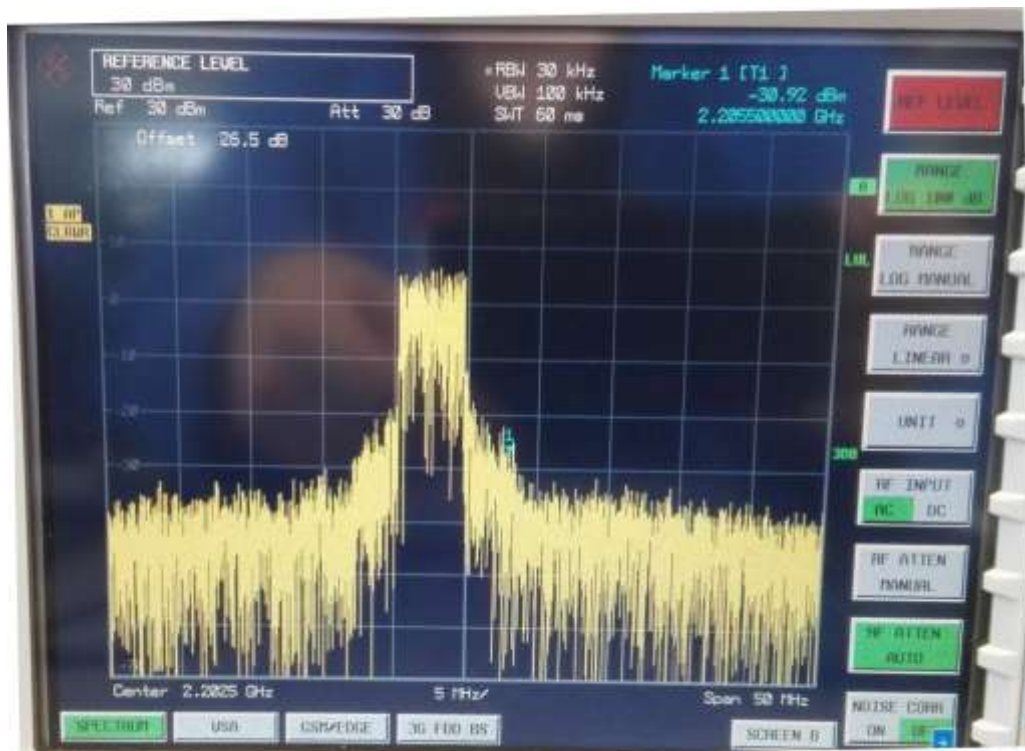


9.3. ábra: Az előtorzító IQ karakterisztikája.

Ha működne az előtorzító akkor a 9.5. ábrán sokkal kisebb lenne a modulált jel kiszélesedése. Számos paraméterét változtattam az előtorzítónak. A memória mélységét, a memória polinom fokszámát, a számításokhoz felhasznált minták számát és az erősítő kivezérlésének mértékét. Az előtorzító egyértelműen hatással van a kimenő jelre, de sajnos nem képes előtorzítani azt úgy, hogy lineárisabb legyen a kimenet.



9.4. ábra: A rendszer kimenete előtorzítás nélkül.



9.5. ábra: A rendszer kimenete előtorzítással.

10. Értékelés, tovább fejlesztési lehetőségek

A diplomatervezés során megismerkedtem a mikrohullámú és nagyfrekvenciás teljesítményerősítők főbb tulajdonságaival, korlátaival. Irodalomkutatásom a különböző analóg és digitális linearizálási technikák széles körét dolgozza fel. Diplomamunkám mélységében foglalkozik a digitális előtorzítás elméletével, annak modellalkotási és matematikai módszereivel. Szimulációkat készítettem az inverz és direkt tanuló architektúra működéséről. A dolgozat második felében ismertettem a modern jelfeldolgozás és rádiózás jelenleg elérhető legfejlettebb eszközeit. Olyan fejlesztő környezeteket, nyelveket és hardvereket használtam, melyek még sok évig használatosak lesznek az iparban. Tovább mélyítettem ismereteimet az RF mérés technika területén és gyakorlati tapasztalatot szereztem digitális modulációk működésével kapcsolatban. Az általam használt áramkörök komplexitása révén számos fontos építőelemet ismerhettem meg, melyek a vezeték nélküli adattovábbításban nélkülözhetetlenek.

A projekt egyértelműen a hibás modellalkotási módszer lecserélésével fejleszthető tovább. Jelenleg a jel I és Q komponenseit teljesen külön kezelem és egymástól függetlenül próbálom előtorzítani őket. A sikertelen előtorzítás oka lehetett az erősítő is. Lehet túl gyors az átmenet a lineáris szakasz és a szaturáció között. Az RF chip sáv szélessége lehet nem elég nagy vagy az általa generált spurious komponensek és zaj jelenléte a kimenő jelben gátolja meg a pontos modellalkotást.

Egy igen sokrétű és több villamosmérnöki tudományterület eredményeit felhasználó gyakorlati munka során digitális előtorzítót terveztem, mely bár nem váltotta be a hozzáfűzött reményeimet, de a feladat nehézségét figyelembe véve így is büszkeséggel tölt el.

Irodalomjegyzék

- [1] Circuitcellar: Amplifier Classes from A to H, <http://circuitcellar.com/cc-blog/amplifier-classes-from-a-to-h/> (2016.04.09.)
- [2] Circuits Today: Class C power amplifiers, <http://www.circuitstoday.com/class-c-power-amplifier> (2016.04.09)
- [3] Foster Fa Dai, Charles Stroud, Dayu Yang, Automatic Linearity and Frequency Response Tests With Built-in Pattern Generator and Analyzer, IEEE transactions on VLSI systems, vol.14 no. 6, 2006 június
- [4] Turusan Kolcuoglu, Linearization of RF Power Amplifiers with Memoryless Baseband Predistortion Method, 2011 május
- [5] e e publishers: Testing the limits of IEEE 802.11ac wireless networking standard <http://www.ee.co.za/article/testing-the-limits-of-ieee-802-11ac-wireless-networking-standardtesting-the-limits-of-ieee-802-11ac.html> (2016.04.10)
- [6] Texas Instruments, Habeeb Ur Rahman Mohammed, What degrades ACPR performance to greater extent: OIP3 or P1dB?, https://e2e.ti.com/blogs_/b/analogwire/archive/2013/04/02/what-degrades-acpr-performance-to-greater-extent-oip3-or-p1db (2016.04.10.)
- [7] Xiaowen Feng, Efficient baseband digital predistortion techniques for linearizing power amplifier by taking into account nonlinear memory effect, <https://hal.archives-ouvertes.fr/tel-01206266/document> (2016.04.10.)
- [8] Maxim Integrated, Predistortion techniques vs. Operation in Back-Off , <https://www.maximintegrated.com/en/products/comms/wireless-rf/rf-predistorters/predistortion-linearization/rfpal-technology/predistortion-techniques-vs-operation-backoff.html> (2016.04.12)
- [9] Mark A. Briffa, Linearization of RF Power Amplifiers, www.geocities.ws/markbriffa/briffa11.pdf (2016.04.12.)
- [10] Abilash Menon, Power Amplifier Linearization Implementation using Field Programmable Gate Array <http://scholarworks.umass.edu/cgi/viewcontent.cgi?article=1099&context=theses> , (2016.04.12.)
- [11] Dr. Wettl Ferenc, Haladó Lineáris Algebra, <http://math.bme.hu/~wettl/okt/linalg/2016/index.shtml> (2016.04.22.)
- [12] Wikipedia: Least Squares Method: https://en.wikipedia.org/wiki/Least_squares (2016.04.28.)

- [13] Dimitry Pelinovsky, McMaster University
<http://dmpeli.math.mcmaster.ca/Matlab/Math4Q3/NumMethods/Lecture2-4.html> (2016.04.28.)
- [14] Sujbert László, Balogh Tibor, Információ feldolgozás laboratórium segédlet - Adaptív szűrők vizsgálata
- [15] Analog Devices, AD9361 Reference Manual
http://www.analog.com/media/en/technical-documentation/user-guides/AD9361_Reference_Manual_UG-570.pdf (2016.10.11.)
- [16] Di Pu, Andrei Cozma, Tom Hill, Four Quick Steps to Production: Using Model-Based Design for Software-Defined Radio
- [17] Xilinx, Zynq-7000 All Programmable SoC
- [18] Xilinx, 7 Series FPGAs Memory Resources
http://www.xilinx.com/support/documentation/user_guides/ug473_7Series_Memory_Resources.pdf (2016.10.12)
- [19] Rhudy Matthew, Brian Bucci, Jeffrey Viperman, Jeffrey Allanach, Bruce Abraham "Microphone Array Analysis Methods Using Cross-Correlations". Proceedings of 2009 ASME International Mechanical Engineering Congress, Lake Buena Vista, (2009)