



M Ű E G Y E T E M 1 7 8 2

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

Hetesi Krisztián

**ANALÓG-DIGITÁLIS ÁTALAKÍTÓK
TESZTELÉSE SZABVÁNYOS
ELJÁRÁSOKKAL**

KONZULENS

Pálfi Vilmos

BUDAPEST, 2017

TARTALOMJEGYZÉK

Összefoglaló	4
Abstract.....	5
1 Bevezetés	6
1.1 A téma megközelítése	6
1.2 Beágyazott rendszerek bemutatása	6
1.3 Jelfeldolgozás.....	6
1.4 Analóg-digitális átalakítókról	8
2 A feladat elméleti háttérének áttekintése.....	12
2.1 Az átalakító gerjesztő jelének megválasztása	12
2.2 Gerjesztő jel paramétereinek meghatározása	12
2.3 A szinuszzellel szemben támasztott követelmények	16
3 A feladatkiírás értelmezése	19
3.1 A feladatkiírás.....	19
3.2 A célprogram specifikációi	19
4 Az önálló munka bemutatása.....	21
4.1 Mérési adatok bevitele	21
4.2 A szinuszjel paramétereinek becslése	22
4.3 A hisztogramteszt	29
4.4 Az átalakító karakterisztikájának meghatározása	33
4.5 Az átalakító hibáinak számítása.....	35
5 Értékelés, saját mérések eredményei	39
6 Összefoglalás.....	47
6.1 Az elkészült feladat értékelése, a fejlesztés lépései	47
6.2 Továbbfejlesztési lehetőségek	48
7 Irodalomjegyzék.....	50

HALLGATÓI NYILATKOZAT

Alulírott Hetesi Krisztián, szigorló hallgató kijelentem, hogy ezt a szakdolgozatot 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. 14.

.....
Hetesi Krisztián

Összefoglaló

A Diplomatervem célja analóg-digitális átalakítókat tesztelő program fejlesztése LabView fejlesztői környezetben. A program képes meghatározni az átalakító komparálási szintjeit, ezek alapján pedig az integrális, és differenciális nemlinearitási hibákat. A komparálási szintek meghatározásához szinuszos gerjesztést alkalmazunk, a pontosabb eredmények érdekében a szinusz paramétereit a tesztelő program képes megbecsülni. Ezeket a paramétereket fel tudjuk használni a hisztogram teszt segítségével a komparálási szintek meghatározásához. Amint ezeket meghatároztuk, megvan az átalakító karakterisztikája, és ekkor az átalakító hibái jellemezhetővé válnak. A fejlesztés LabView környezetben zajlik. A diplomamunka készítése alatt megismerkedhettem a fejlesztői környezettel, gyakorlati feladatok elvégzése közben. A program kifejlesztése után egy valóságos átalakító tesztelése is feladata a diplomamunkámnak. Miután a végleges tesztprogram elkészül, és hibátlanul működik, felkerül a világhálóra, hogy azt kutatási, oktatási, vagy egyéb feladatokra fel tudják használni.

Abstract

My Thesis is about a testing program for analog-to-digital converters in LabView development environment. The program is able to identify the transfer characteristic of the ADC by the estimation of the transition levels. Once the transfer characteristic is known, the error of the converter can be described using the standard quantities. The transition levels are determined using the sinewave histogram test. In this method, the ADC is excited with a sine wave input, then a histogram is created from the digital codes. Accurate estimation of the transition levels requires precise information about the sine parameters, so the program is able to estimate amplitude, frequency, initial phase and DC offset component of the input signal. Once these steps are done, the offset and gain error, and the nonlinearity errors of the ADC can be evaluated. The development took place in LabView environment. I have learnt the development environment during the preparation of the thesis by solving practical problems with it. After the development and testing with simulated data was done, I have tested the program with real measurements. The program is still under development, once its final version is ready, it will be published for research and educational purposes.

1 Bevezetés

1.1 A téma megközelítése

A diplomatervem egy analóg - digitális átalakítókat tesztelő program fejlesztése. Ezen átalakítóknak leginkább a jelfeldolgozás folyamatában, beágyazott rendszerekben veszik hasznukat. A szakdolgozat témám ismertetése előtt, a bevezetés rovat felelős azért, hogy az olvasó egy kielégítő képet kapjon a témakörrel általánosságban, illetve megismerje az elkészültének motivációját.

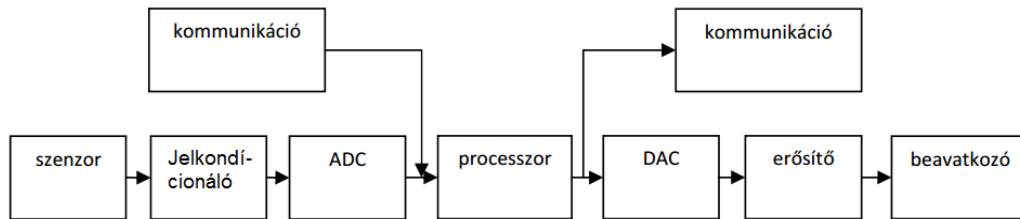
1.2 Beágyazott rendszerek bemutatása

A beágyazott rendszerek az információs technológia egyik részterületét képviselik. Az általános számítógépekkel ellentétben általában egy adott célfeladatot látnak el, amelyre optimalizálják. Ennek következtében alacsony költséggel, mérettel, illetve magas megbízhatósággal rendelkeznek. A legtöbb esetben a fizikai környezettel is kapcsolatban állnak, szenzorok és beavatkozók segítségével. Jellegzetes példája az ilyen rendszereknek a személygépkocsi elektronikája.

Vegyük például az ABS-t. Az ABS rendszer azért felelős, hogy a kerekek lehetőleg ne kerüljenek álló helyzetbe, mielőtt azt az autó tenné. Ez az egység egy szenzor segítségével érzékeli a kerekek forgási sebességét, miközben fékezünk. Az információt továbbítja a beágyazott rendszer „agyába”, egy processzorba, mely a szükséges műveletek elvégzése után továbbítja a döntések eredményét a beavatkozónak, mely cselekszik. A beavatkozó vagy engedi, hogy a lábunk által benyomott fékpedállal a fék mechanizmus dolgozzon, vagy pedig nem engedi, hogy az általunk választott fékező erő eljusson a fékekig, ezáltal megakadályozva, hogy állóra fékezze a kereket.

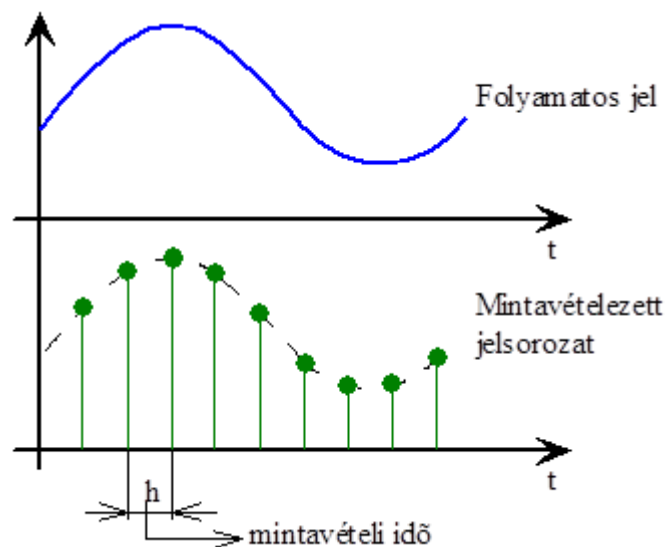
1.3 Jelfeldolgozás

A digitális jelfeldolgozás (*angolul Digital Signal Processing, röviden DSP*) egy eljárás, folyamat melyet a beágyazott rendszer segítségével végzünk.



1. ábra. A jelfeldolgozás általános blokkvázlata

Az ábrán láthatjuk a jelfeldolgozás folyamatát. Az ilyen rendszer egy szenzorral kezdődik, mely egy fizikai jelet, jelenséget egy mérhető villamos jellé alakít. Ezek a szenzorok általános esetben analóg jeleket bocsátanak ki magukból, mely az adott fizikai jelenség mértékének függvényében vesz fel értéket. Ezt az analóg jelet gyakran kell digitalizálás előtt kezelni. Például, ha parányi a jel, akkor erősítőt alkalmazunk, vagy ha a zajterjedést akarjuk megállítani, akkor galvanikus leválasztást. A teljesség igénye nélkül csak két példát írtam a jelek kondicionálására, ezek mellett többféle eljárás létezik. Megfelelő igazítások után a jel készenáll a digitalizálásra, a szakdolgozat témám tárgyát képző átalakító fogadja az analóg jelet. A digitalizálás után egy mintavételezett, kvantált jelet kapunk. A mintavételezés azt jelenti, hogy bizonyos időközökben, úgynevezett mintavételi időnként felírjuk jel erősségét, értékét.



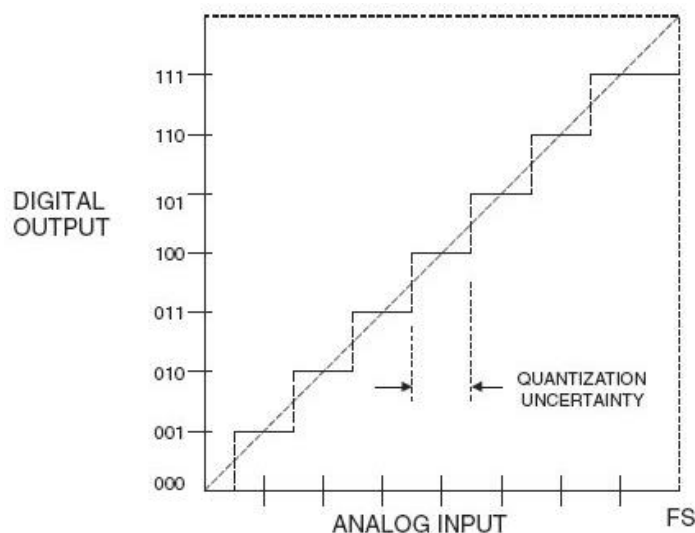
2. ábra. Mintavételezés

A kvantálás azt jelenti, hogy a mintavételezett értékeket szabályszerűen beskálázott szintekhez közelítjük. Mivel használtam azt a szót, hogy közelítés, látható, hogy itt már módosítunk az eredeti jelen, torzítjuk. Ebből ered a kvantálási zaj.

A digitalizált jel végül a rendszer processzorába kerül, ahol az adott felhasználási mód szerint számítások mennek végbe. A jelfeldolgozás esetében úgynevezett jelprocesszor végzi a számításokat, mely nagyobb pontosságú számokkal képes dolgozni és gyorsabban, az architektúrája és az utasításkészlete miatt. Egy beágyazott rendszer esetében a processzor akár több forrásból is kaphat információkat, valamint küldhet több egység felé is. Ezért látható az ábrán, hogy kommunikációra is képes.

1.4 Analóg-digitális átalakítókról

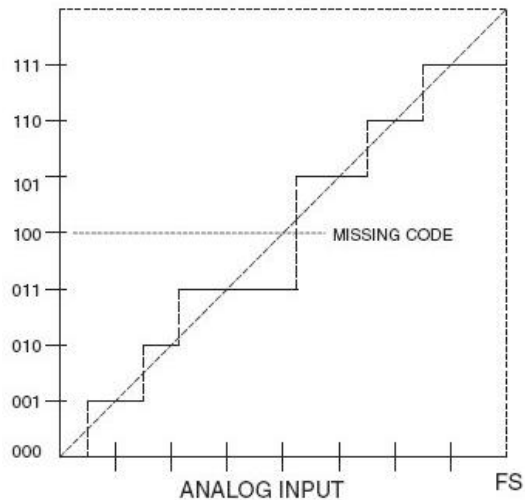
Az analóg-digitális átalakítók szerepe jelentős a beágyazott rendszerek és a jelfeldolgozás világában. Az átalakítók feladata, hogy egy analóg jelet, információt, átalakítsanak egy mintavételezett, kvantált jellé. Egy ideális átalakító karakterisztikája az alábbi ábrán látható:



3. ábra. Az ideális átalakító karakterisztikája

Az átalakító minősége, pontossága nagyban hozzájárul az adott beágyazott rendszer pontosságához. Az előbbi ábrán egy ideális karakterisztika lett felvázolva, a valóságban technológiai okokból, ezt csak közelíteni lehet (4. ábra). Szükség van valamilyen

mérőszámokra, amivel jellemezni lehet egy átalakítót, és a hibáinak a mértékét is, a fizikai kialakításától függetlenül. Ezeket a későbbiekben fogom ismertetni.

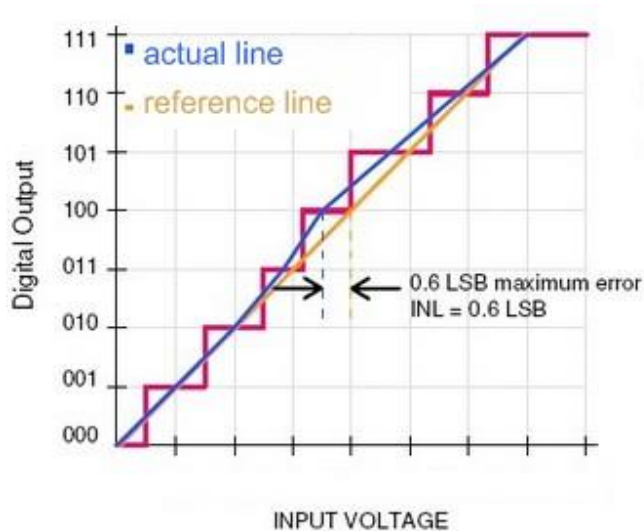


4. ábra. Nem ideális átalakító karakterisztikája [7]

Sok esetben elvárás, hogy az átalakítók nagy pontossággal rendelkezzenek, viszont előfordulhat, hogy nem egyértelmű a termékek specifikációja. Ezért van szükség egy eljárásra, ami független az átalakító architektúrájától, és a technológiai kialakításától. Ezen eljárás legyen képes egy olyan teszt végrehajtására, ami eredményként közli velünk az átalakító hibáit.

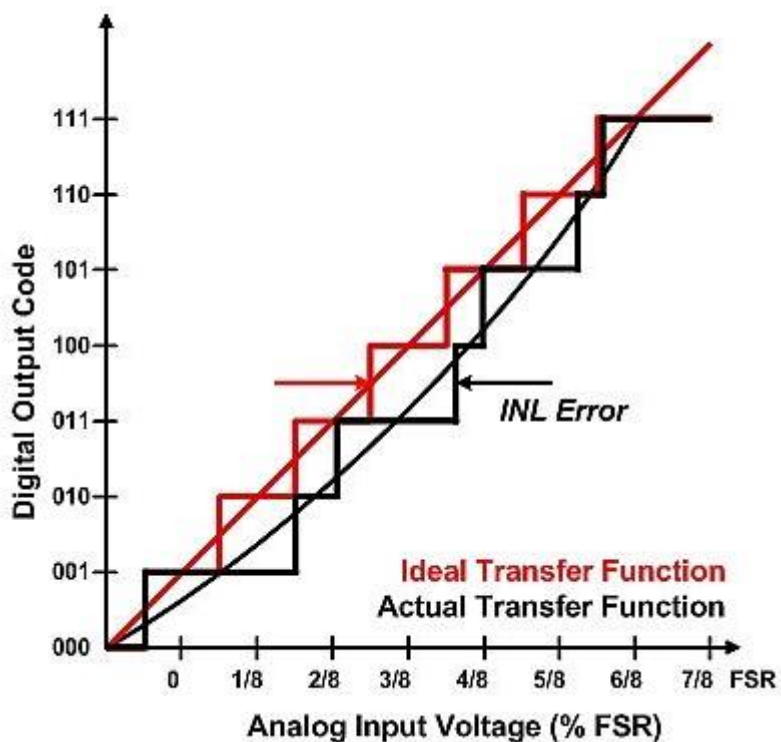
Az átalakítót jellemző mérőszámok:

- **Komparálási szint:** a bemeneti tartományt ezek a szintek 2^b szintre osztják fel. A jelölése: T_k , ami a k-adik komparálási szintet jelenti.
- **Kvantumnagyság (LSB):** az ideális átalakító bármely két szomszédos komparálási szintje közötti távolság.
- **Karakterisztika:** Az imént már bemutatásra került, a kimenet és a bemenet között lévő összefüggést adja meg. Ideális esetben egy lépcsőfüggvény.
- **Referencia egyenes:** illesztett egyenes a komparálási szintek pontjaira, végpontokra illesztve, vagy az ún. „Least-squares” eljárással.



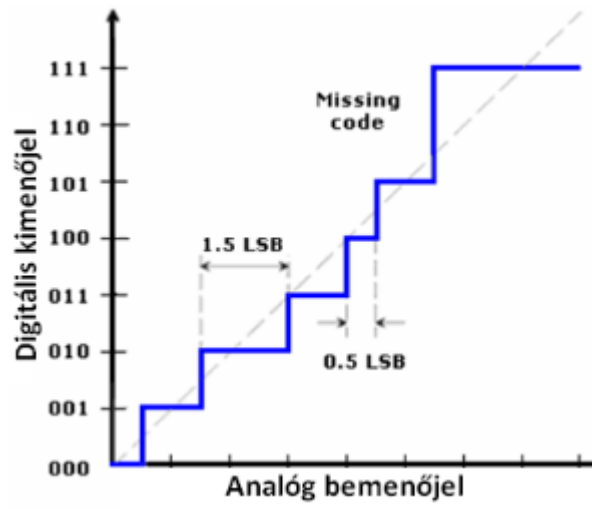
5. ábra. Referencia egyenes [2]

- **Erősítés:** a referencia egyenes meredeksége, ideális esetben LSB.
- **Ofszet:** a referencia egyenes állandó összetevője, ideális esetben $LSB/2$.
- **Integrális nemlinearitás (INL):** a komparálási szintek távolsága a referencia egyenes pontjaitól.



6. ábra. Integrális nemlinearitás

- **Differenciális nemlinearitás (DNL):** a kódszélességek eltérése LSB-től.



7. ábra. Differenciális nemlinearitás

2 A feladat elméleti háttérének áttekintése

2.1 Az átalakító gerjesztő jelének megválasztása

Az átalakítót gerjesztő jel kiválasztása fontos tényező a tesztelés végeredményének szempontjából. Az alapvető jelek a háromszög jel, az exponenciális jel és a szinusz jel. Alacsony frekvencia esetén háromszögjel segítségével akár közvetlenül is meg lehetne mérni az átalakító karakterisztikáját. Azonban a hibamentes előállítása nehézkes, és az átváltási tüskék torzítják a sűrűség függvényt. A háromszögjelnél kevésbé elterjedt az exponenciális jel. Ennek a jeltípusnak az előnye, hogy nagyon könnyen előállítható egy kondenzátor kisütésével egy ismert ellenálláson, továbbá három paraméter segítségével leírható. Hátránya viszont, hogy kis feszültségeken a jel meredekségének csökkenése miatt érzékeny a zajokra.

A harmadik típus a szinuszjel. Ennek a jeltípusnak a zajmentes előállítása kevésbé ütközik akadályokba, mint a többi jel esetében. Az átalakító karakterisztikájának meghatározása kevésbé triviális feladat, mint a háromszögjel esetében, viszont a hisztogramteszt segítségével ez a probléma áthidalható. Így az átalakító karakterisztikájának, hibáinak meghatározására a szinuszjelre esett a választás ebben a projektben.

2.2 Gerjesztő jel paramétereinek meghatározása

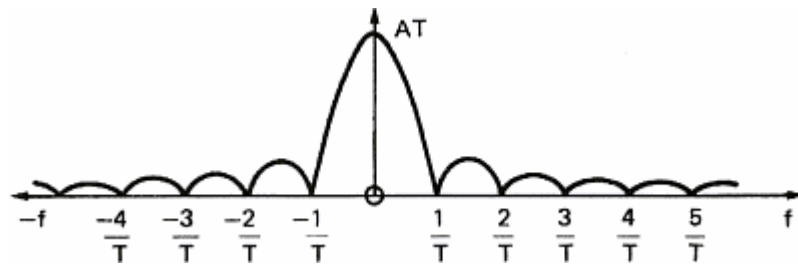
Az átalakító karakterisztikájának meghatározásához szükségünk van a szinuszjel paramétereire. A tervezett program kezelése nehézkes volna, ha a felhasználónak minden esetben ismernie kellene az átalakított jel paramétereit, hogy azt bemenetként megadhassa a tesztelő program számára. A jelgenerátorok bizonytalansága miatt a paraméterek értékeit általában nem is ismerjük kellő pontossággal. Emiatt szükséges valamilyen eljárás, amely megbecsüli ezen paramétereket. Létezik többféle, szabványos és nem szabványos eljárás is a feladat elvégzésére [6]. A szinuszjel paramétereinek meghatározásához az alábbi jelmodellt alkalmazom a továbbiakban:

$$x[k] = C + A \cos\left(\frac{2\pi * J * k}{N}\right) + B \sin\left(\frac{2\pi * J * k}{N}\right) \quad (1)$$

Mivel a tesztet véges számú mintával végezzük, ezt úgy tudjuk modellezni, mintha a jel meg lenne szorozva egy négyszögablakkal. A négyszögablak az alábbi módon néz ki:

$$\text{Időtartományban: } f(t) = \begin{cases} 1 & \text{ha } T > t > 0 \\ 0 & \text{egyébként} \end{cases} \quad (2)$$

$$\text{Frekvenciatartományban: } X(f) = \left| \frac{\sin(f)}{f} \right| = |\text{sinc}(f)| \quad (3)$$



8. ábra. Négyszögjel frekvenciatartományban [4]

Látható, hogy a jel frekvenciájánál nem torzít a jelen, amennyiben az egy tiszta szinusz jel volt. Viszont ha nem volt koherens a mintavételezés, akkor ott is megjelennek komponensek, ahol nem várnánk, ez a spektrumszivargás jelensége. Emellett az a komponens, amit becsülni szeretnénk, csillapítva lesz, ez pedig a csúcs letörés jelensége.

Megvizsgáltam továbbá a Flat top ablakot is. Ehhez az alábbi jellel kell szorozni:

$$w(t) = \sum_{k=0}^4 (-1)^k * a_k * \cos\left(\frac{k * 2\pi * t}{T}\right) \quad (4)$$

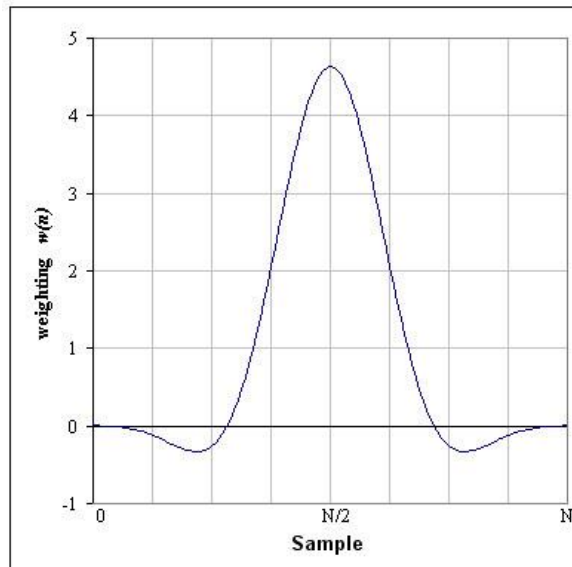
$$a_0 = 0.215578948$$

$$a_1 = 0.416631580$$

$$a_2 = 0.277263158$$

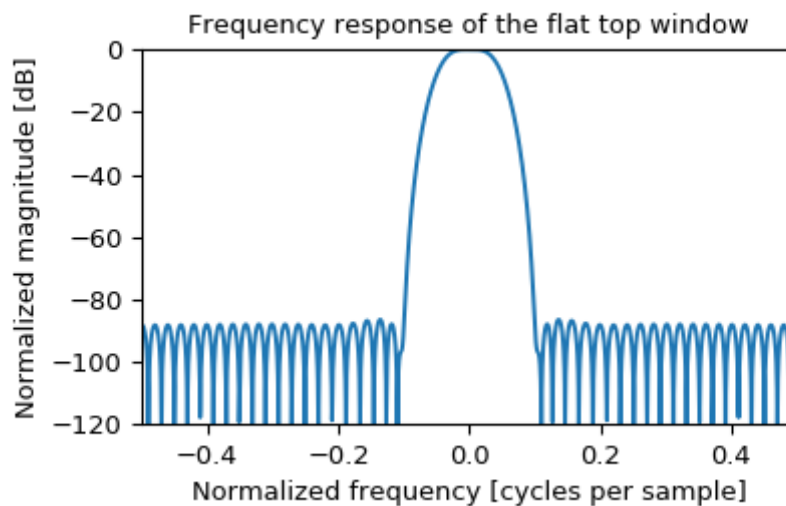
$$a_3 = 0.083578947$$

$$a_4 = 0.006947368$$



9. ábra. Flat top ablak időtartományban

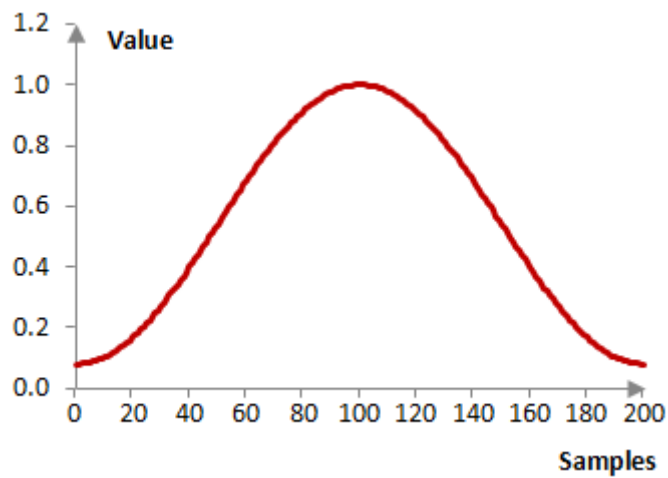
Ennek az ablaknak a frekvenciatartománybeli képe:



10. ábra. Flat top ablak frekvenciatartományban

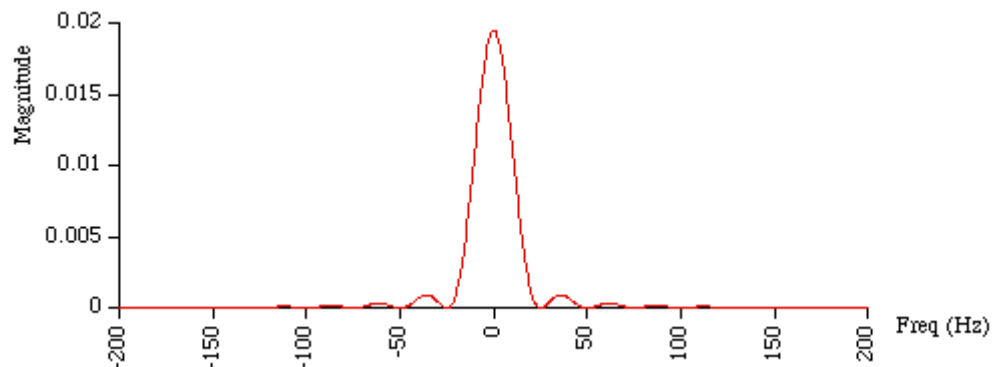
Az ábra megmutatja, hogy a szinuszjel frekvenciájánál tágabb sávszélességen belül nem romlik a jel amplitúdója, jobban megoldja a csúcsletörés problémáját, így ez az ablak alkalmasabb az amplitúdó meghatározásra. A másik időablak, amit megvizsgáltam, a Hanning ablak, melynek időtartománybeli alakja egy haranggörbével jellemezhető:

$$w(t) = 0.5 - 0.5 * \cos\left(\frac{2\pi * t}{T}\right) \quad (5)$$



11. ábra. Hanning ablak időfüggvénye

Frekvenciatartományban pedig:



12. ábra. Hanning ablak frekvenciatartományban

Észrevehető, hogy ezt az ablakfüggvényt alkalmazva a frekvenciatartományban a jel frekvenciáján kívül sokkal gyengébb az erősítés, mint a többi ablaknál. Ez tehát jól megoldja a spektrumszivárgás jelenségét. Így frekvenciabecsléshez kiválóan alkalmazható.

A nem szabványos eljárások közül az interpolált FFT használatával végzett eljárást vizsgáltam munkám során. Az eljárás előnye, hogy nagyon gyors az iteratív eljárásokhoz képest, bár az eredmények jósága alulmarad nem koherensen mintavételezett jelek esetében. Ilyenkor az becslés eredményét korigálni kell, a frekvenciabecsléshez szükséges korrekciót adta meg Ming és Kang a cikkükben [5],

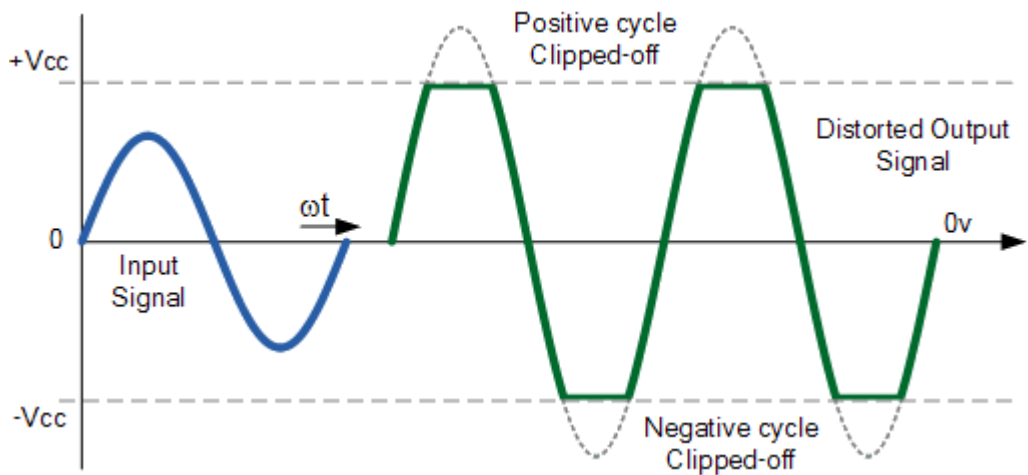
ablakozatlan, illetve Hanning ablakot használva. Hanning ablakot használva, a korrekciókat elvégezve elég pontos eredményt ad a szinuszjel frekvenciájára.

A szabványos eljárások közül kettőt jártam körül a szakdolgozat témám irodalomkutatása közben. Az egyik a háromparaméteres, a másik a négyparaméteres szinuszbecslés [1]. A háromparaméteres egyszerűbb, gyorsabb eljárás, nem tartalmaz iterációt. A korábban leírt jelmodell képletből (1), az A, B, C paramétereket tudja megbecsülni zárt alakban. A négyparaméteres becslés ezzel ellentétben J -t, azaz a periódusok is számát képes meghatározni. Ez már egy iterációt tartalmazó módszer, aminek szüksége van egy megfelelően pontos J_0 kezdőértékre is. A munkámban a maximális pontosság elérése érdekében kombináltam az IpFFT-s eljárást a négyparaméteres becselővel. A becselő kezdeti J_0 értékét a Hanning ablakozott jel IpFFT eljárásának eredménye adja.

2.3 A szinuszjellel szemben támasztott követelmények

A korábban említett eljárásokban elhangzott a koherens mintavételezés fontossága. Egy mintavételezés akkor koherens, ha teljesül következő feltétel: a mintavételezés során a jelből egész számú periódust mérünk. A későbbiekben bemutatott eljárások miatt célszerű még egy feltételnek megfelelni: hogy a mintavételezett minták száma, és a periódusok száma relatív prímek legyenek.

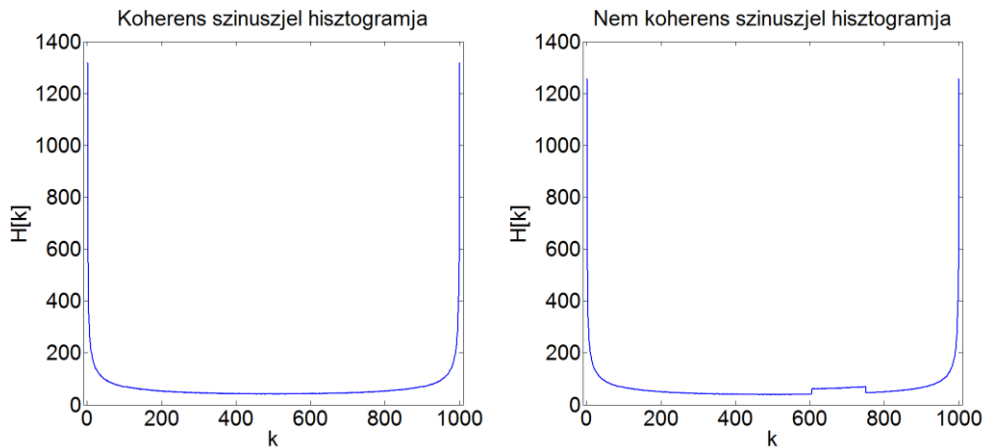
A szinuszjel paramétereinek becslésének folyamatához is szükséges a koherens mintavételezés, viszont az interpolált FFT-s becslés utáni korrekció érzéketlenné tudja tenni erre a problémára. Amit viszont ezek az eljárások nem tudnak kezelni, az az, hogy mi történik akkor, ha túlvezéreljük az átalakítót. Ilyenkor a becselő nem szinusz jelet lát, hanem egy eltorzult periodikus jelet:



13. ábra. Túlvezérlés hatása [3]

Ilyenkor a becselő nem tudja meghatározni a jel paramétereit helyesen, mivel a túlvezérlés miatt felharmonikusok jelennek meg a frekvenciatartományban. Mivel a frekvencia meghatározását jó elnyomású ablak alkalmazása esetén ez nem befolyásolja, így a javítást a négyparaméteres becslés előtt elég elvégezni. A megoldás az lehet, ha a négyparaméteres eljárást nem a teljes mintasorozatunkra futtatjuk le, hanem kivágjuk belőle a túlvezérelt mintákat. Így a becslés jó pontossággal képes meghatározni annak a szinuszjelnek a paramétereit, melynek a csúcsai levágódtak.

A hisztogramteszt folyamata érzékeny mind a túlvezérlés, mind a nem koherens mintavételezés problémájára. A túlvezérlés problémáját viszonylag könnyen lehet orvosolni. Ha a jel sűrűségfüggvényéből elhagyjuk a két kiugró szélét, a két leggyakoribb mintát, nagyobb pontosságot érhetünk el. Továbbá, ha a kumulatív hisztogram értékei közül elhagyjuk a legkisebb, és a legnagyobb 10%-ba tartozó mintákat, érzéketlenné tettük a karakterisztika meghatározását a túlvezérlésre. Cserébe fel kellett áldozunk mintákat, és a karakterisztika nem kerül meghatározásra a teljes kódterületen.



14. ábra Koherens mintavételezés histogramra gyakorolt hatásása [6]

A histogramteszt érzékeny arra, ha nem egész számú periódusunk van a mintasorozatunkban. Ezt könnyű belátni, mivel ha egész számú periódust mintavételezünk, majd hozzáveszünk még egy egész periódust, akkor arányosan minden kódhoz hozzáadódnak az extra histogram értékek. Miközben, ha például, egy fél periódust adnánk hozzá az egészhez, akkor csak a kódok felénél növekednének meg a histogram értékek. Ebből következik, hogy a kumulatív histogramunk sem azokat az értéket veszi fel, mint amiket koherens esetben. Ez esetben pedig a komparálási szintek lesznek rosszul meghatározva.

A nem relatív prím feltétel azt jelenti, hogy a mintavételezett jelsorozat periódus száma és a mintaszáma relatív prímek. Tehát a legnagyobb közös osztójuk legyen 1. Ha ez teljesül, akkor minden periódusban a mintavételezett pontok különböző helyen helyezkednek el fázisban, tehát több információt kapunk az átalakító komparálási szintjeiről. Ezáltal a histogramteszt varianciája csökken, pontosabban tudjuk meghatározni a komparálási szinteket.

A nem koherens mintavételezést teljesen megoldó algoritmust nem találtam kutatásaim során, viszont a periódusszámot egészé téve a legtöbb esetben megoldást jelenthet. A legegyszerűbb, számításokban is csekély eljárás az lehet, ha szimplán eldobunk egy szakasznyi mintát a sorozatból, pont annyit, hogy egész számú periódusnyi minta maradjon. Ezzel információt veszítünk, viszont általában nagy mintaszámú méréseket készítünk, így elhanyagolhatóan kis veszteség az egynél kisebb periódusnyi mintaszám.

3 A feladatkírás értelmezése

3.1 A feladatkírás

A korábbi fejezetekben ismertette lett a feladat motivációja. Az A/D-konverterek felhasználás előtti ellenőrzése, tesztelése, az adatlapon való egyértelmű specifikálása mind nehéz feladat. Sok a bizonytalanság, az eltérő értelmezés. Ezért született meg az IEEE 1241-2010 dokumentum, amely definiálja az egységes terminológiát és eljárásokat. A legfontosabb eljárás alapja a szinuszillesztés: egy jól definiált algoritmus (legkisebb négyzetek módszere, LS) segítségével meghatározzuk az ADC kimeneti mintáira legjobban illeszkedő szinusz jelet, majd a mintáknak és a szinuszjelnek a különbsége megadja a konverter hibáját, mely tovább analizálható. Az LS módszer általában nagyon jó eredményeket ad, de bizonyos esetekben (pl. túlvezérlés) a becslő jósága szignifikánsan romlik.

3.2 A célprogram specifikációi

A tesztprogram fejlesztése LabView környezetben történt. Grafikus felületének köszönhetően a fejlesztés gyorsan tanulható, továbbá jól strukturált projekteket lehet létrehozni a „*sub vi*”-k segítségével. A „*vi*”-k a LabView rendszerében a függvények, melyek külön-külön saját fájlokban foglalnak helyet, a „*sub vi*” jelentése pedig a „*vi*”-ba iktatott másik „*vi*”. Ez a környezet kifejezetten alkalmas jelfeldolgozási feladatok megoldására.

A cél az, hogy a program képes legyen egy átalakítót tesztelni, mindössze egy, az átalakító kimenetét képező adatsorozatból. Amennyiben rendelkezésünkre áll „*tdms*” formátumban az A/D kimenete, a tesztprogram azt fel tudja használni. A program egy szinusz jel digitális átalakítását igényli, mivel a választott szabványos eljárások, algoritmusok a szinuszjel paramétereivel képesek számolni.

A program az egyes részfeladatok megoldását is képes monitorozni, nem csupán a fájl bemenete után írja ki az átalakító hibáit. A bemeneti fájl kiválasztása után, a program megjeleníti a bemeneti jelet, és a paraméterek becsült értékét is meghatározza. Ez abban az esetben is hasznos lehet, ha valamiért nem ismerjük a bemeneti állományban lévő jelet, vagy akár ellenőrzésre is használható. Megjeleníti a jel időfüggvényét,

meghatározza az amplitúdót, ofszetet, a kezdőfázist, továbbá a mintában lévő periódusok számát, mely mennyiség arányos a jelfrekvenciával .

A program a hisztogramtesztet is végrehajtja, így a minták hisztogram ábráit is képes kirajzolni. A periódusok számának becslésén túl a hisztogramteszt ábrái is árulkodnak arról, ha nem koherens a mintavételezés, így a felhasználó orvosolni tudja a tesztelés pontatlanságát azzal, ha új mintasorozatot hoz létre, figyelve a koherens mintavételezés szabályainak betartására.

A hisztogram ábrái mellett a program képes kirajzolni az átalakító karakterisztikáját. Mint ahogy a bevezetésben erről szó esett, az ideális átalakító karakterisztikája egy LSB meredekségű egyenesre illeszkedő lépcsős függvény. A karakterisztika eltorzulása is kimutatja már az átalakító hibáit, viszont a már fentebb is említett nem koherens mintavételezést is jelentheti. Emiatt a karakterisztika kirajzolása fontos pontja a program specifikációinak.

A nemlinearitási hibák meghatározása szintén célja a programnak. Ezen értékek kijelzésének fontossága triviális. Mind az integrális nemlinearitás, mind a differenciális nemlinearitás értékeinek kirajzolása sokkal felhasználó barátibb módszer, mintha táblázatban adna a program visszajelzést. Az INL és DNL maximális értékét viszont számként is jelzi a program.

4 Az önálló munka bemutatása

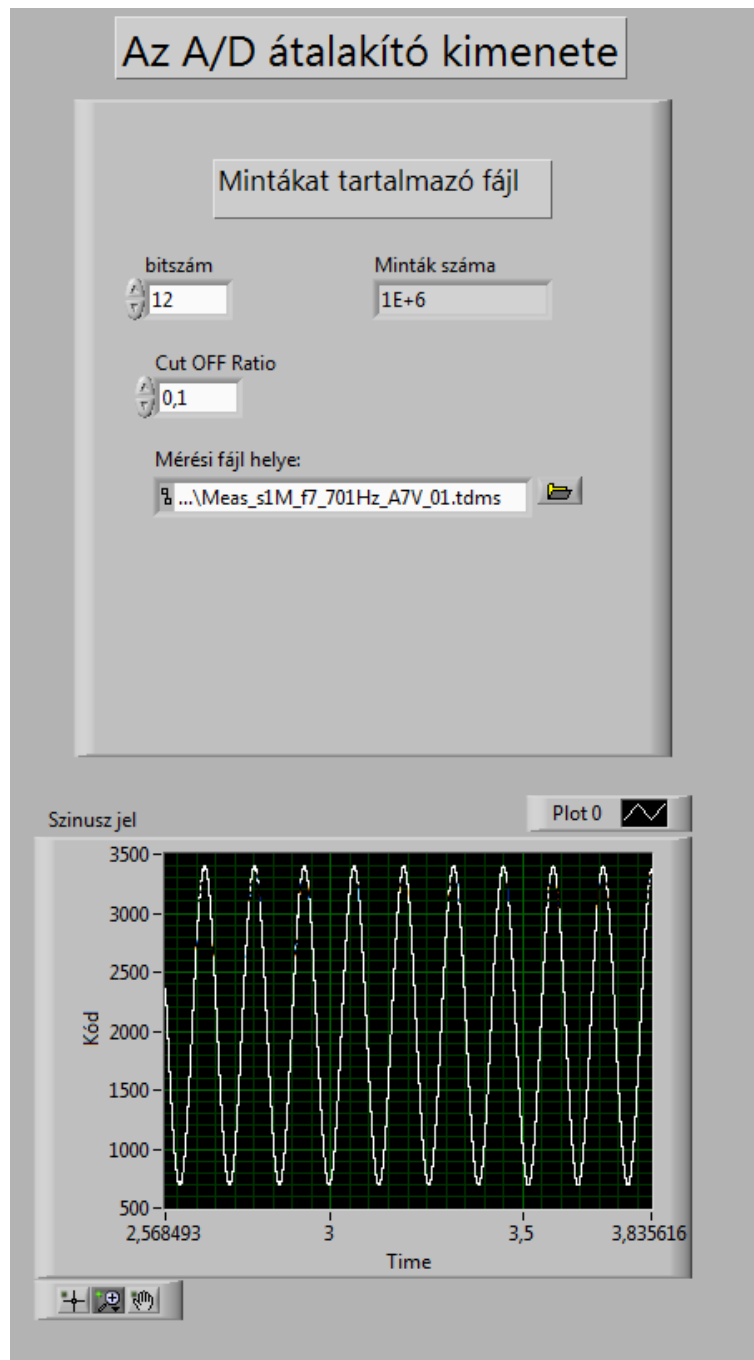
4.1 Mérési adatok bevitele

A program ezen része az, ahol a felhasználó hozzákezdhet az átalakító teszteléséhez. Itt lehet betallózni a mérésből származó mintákat tartalmazó fájlt. A helyes működés miatt meg kell adni a tesztelendő átalakító bitszámát. Ha nagyobb bitszámot adunk meg, mint a tesztelni kívánt A/D valódi bitszáma, az nem lesz hatással a működésre az eredmények szempontjából, viszont több ideig fut a tesztelési folyamat. Azonban, ha kisebb bitszámot adunk meg, kisebb kódtartományon fog dolgozni a program, ezáltal helytelen eredményekre jutunk. A helyes bitszám beírása a felhasználó felelőssége.

A bitszám mellett szintén a felhasználó feladata az úgynevezett „cut-off ratio” értékének megadása, melynek funkcióját majd a későbbi fejezetekben kívánom bemutatni. Ezt az értéket célszerűen 0,1-nek érdemes megválasztani, hogy kellő mértékű korrigálást tudjon végezni a hisztogram szélein, viszont ne dobjon el túl sok információt az adathalmazból. Ennek a funkciónak a bemutatását a későbbiekben részletesebben elvégzem. Ez az ablak kijelzi, hogy a mintákat tartalmazó fájl hány darab mintát tartalmaz. Ez önellenőrzési célokat is szolgáltat, amennyiben a felhasználó hozta létre korábban a tesztelni kívánt mérési fájlt.

A program ezen a blokkon tartalmaz egy „kijelző” modult is, mely kirajzolja a mintasorozat pontjait. Ezzel a funkcióval lehet ellenőrizni, hogy a kívánt mérési fájl lett-e betallózva. Erről a kijelzőről az alapvető jelparaméterek akár szabad szemmel is leolvashatók, az amplitúdó, ofszet, vagy akár a periódusok száma is.

A program, ahogy az már korábban említve lett, „tdms” fájlokat tud kezelni, beolvasni. A választott eljárások, algoritmusok miatt, a fájl tartalmának mintavételezett, kvantált szinuszelnek kell lennie. A mintasorozatra lehetőleg teljesülnie kell a koherens mintavételezés előírásainak, miszerint csak egész számú periódus legyen, és teljesüljön továbbá a relatív prím feltétel a minták száma és a periódusok száma között. Ha nem teljesül az előbbi, akkor az esetek többségében - a későbbiekben bemutatott eljárásoknak köszönhetően - a program ezt a problémát át tudja hidalni.



15. ábra. Bemenet beállításának felhasználói felülete

4.2 A szinuszjel paramétereinek becslése

Mint már említettem korábban, a hisztogram teszthez, és a többi folyamathoz a szinusz paramétereinek ismerete szükséges. Mivel az átalakító kimenetén csak egy kódsorozatot látunk, így ebből a kódsorozatból kell meghatároznunk a szinusz jel paramétereit. Először a mintavételezett jelben található periódusok számát kell megbecsülni. Erre egy nem szabványos eljárást használok, interpolált FFT segítségével

állapítom meg a paraméter értékét. A dolgozat korábbi fejezetében (2.2), bemutattam több ablakfüggvényt is.

A nem koherens mintavételezés kezelésére a jelet Hanning ablakkal szorzom az FFT végrehajtása előtt. A Hanning ablak jól csillapítja a szivárgást és a letörést is, viszont amplitúdót pontatlanul lehet vele meghatározni. Jelen esetben viszont a periódusszám meghatározásra használom, arra elegendően pontos. Hanning ablakhoz az alábbi jelsorozattal kell szorozni az átalakítóból kijövő jelet:

$$W_{Hanning}[k] = 0.5 - 0.5 * \cos\left(\frac{2\pi * k}{N}\right) \quad (6)$$

A periódus szám meghatározáshoz felhasználjuk, hogy az az alábbi alakban áll elő:

$$J = K + \Delta K \quad (7)$$

Itt K egész szám, ΔK pedig egy tört szám, amely $[-0.5 ; 0.5]$ intervallumon helyezkedik el.

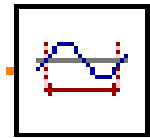
K számításához felhasználjuk a mintasorozat FFT-jét. Az FFT eredményéből csak az $[1; N/2]$ intervallumot tartjuk meg. Erre azért van szükség, mert a maximális értéket használjuk fel a becsléshez, a nulladik index, az egyen komponens pedig lehet nagyobb, mint a periódusszám komponense. Az $N/2$ -es felső határra valós jelek komplex konjugált szimmetriája miatt van lehetőség. Tehát a K kiszámításának módja:

$$K-1 = \max \left| \text{FFT}(X) \left[1: \frac{N}{2} \right] \right| \quad (8)$$

A ΔK kiszámításának módja pedig figyelembe veszi az FFT maximális értékének a szomszédjait is.

$$\Delta K = \begin{pmatrix} \frac{2Y_{k+1} - Y_k}{Y_{k+1} + Y_k} & \text{ha: } Y_{k+1} \geq Y_{k-1} \\ \frac{Y_k - 2Y_{k-1}}{Y_{k-1} + Y_k} & \text{ha: } Y_{k+1} < Y_{k-1} \end{pmatrix} \quad (9)$$

Itt $Y = |FFT(X)|$, azaz a Fourier transzformált abszolút értéke.



16. ábra Periódusok számának becslő "sub-vi"-a

A program forráskódjában a periódusok számának meghatározásáért külön sub vi felel, melynek az ikonja az ábrán látható. A „doboz” bemenetként csupán a jelsorozatot kapja, kimenetei pedig a becsült periódusok száma, a „delta” korrekciós szám, és az FFT kijelzése. Itt történik a Hanning ablak használata, az FFT végrehajtása, illetve a fentebb bemutatott módszerrel a periódus szám meghatározása. A LabView tartalmaz beépített FFT „vi” egységeket, így ezt nem kellett tervezéskor implementálni.

A szinusz jel többi paraméterének, és a periódusok számának pontosabb meghatározásához a szabványos 4 paraméteres becslést választottam. Ehhez az alábbi alakban kell figyelembe venni a jelet:

$$x[k] = C + A \cos\left(\frac{2\pi * J * k}{N}\right) + B \sin\left(\frac{2\pi * J * k}{N}\right) \quad (10)$$

A fenti kifejezésben J a becsült periódus szám, N pedig a mintaszám. Az eljárás az A, B, C és J paramétert becsli meg, iteratív módon. A számítások megértéséhez feltétlenül szükséges a pszeudo inverz kifejezés ismerete. A kiszámítási módja:

$$X^\dagger = (X^T * X)^{-1} * X^T \quad (11)$$

A fenti kifejezésben X^T az X mátrix transzponált mátrixa. A LabView-s megvalósításban viszont nem feltétlenül kell ismerni, mivel beépített funkció blokk áll rendelkezésre a pszeudo inverz mátrix számítására.

Én tapasztalati úton az elkészült programban 20 iterációt választottam, a kívánt pontosság eléréséhez. Az alábbi módon kaphatjuk meg az értékeket az i . iterációban:

$$P_i = \begin{bmatrix} A_i \\ B_i \\ C_i \\ \Delta J_i \end{bmatrix} = [D_i]^\dagger * X \quad (12)$$

Itt a **D** mátrix definíciója az alábbi:

$$D_i = \begin{bmatrix} \frac{\partial k}{\partial A} & \frac{\partial k}{\partial B} & \frac{\partial k}{\partial C} & \frac{\partial k}{\partial J} \end{bmatrix} \quad (13)$$

Vagyis:

$$D_{i+1} = \begin{bmatrix} \cos\left(\frac{2t(0)\pi J_i}{N}\right) & \sin\left(\frac{2t(0)\pi J_i}{N}\right) & 1 & \frac{A_i 2t(0)\pi}{N} \sin\left(\frac{2t(0)\pi J_i}{N}\right) + \frac{B_i 2t(0)\pi}{N} \cos\left(\frac{2t(0)\pi J_i}{N}\right) \\ \cos\left(\frac{2t(1)\pi J_i}{N}\right) & \sin\left(\frac{2t(1)\pi J_i}{N}\right) & 1 & \frac{A_i 2t(1)\pi}{N} \sin\left(\frac{2t(1)\pi J_i}{N}\right) + \frac{B_i 2t(1)\pi}{N} \cos\left(\frac{2t(1)\pi J_i}{N}\right) \\ \vdots & \vdots & \vdots & \vdots \\ \cos\left(\frac{2t(n)\pi J_i}{N}\right) & \sin\left(\frac{2t(n)\pi J_i}{N}\right) & 1 & \frac{A_i 2t(n)\pi}{N} \sin\left(\frac{2t(n)\pi J_i}{N}\right) + \frac{B_i 2t(n)\pi}{N} \cos\left(\frac{2t(n)\pi J_i}{N}\right) \end{bmatrix} \quad (14)$$

A becslés első iterációjában $A_0, B_0 = 0, J_0$ pedig az interpolált FFT-vel kiszámolt periódus szám. A $t(k)$ jelöli az időtömböt, mely a minták sorszámaiból áll, alap esetben 0-tól a minták darabszámaig egyesével növekedő számokból áll, tehát 1-től $N-1$ -ig. A túlvezérlés okozta hibák kiküszöbölése érdekében van egy egység, mely ezt kezeli, erre a későbbiekben kívánok kitérni.

Látható hogy a fenti számítás még nem adja meg J_i -t, hanem csak ΔJ_i -t. Így a kiszámítási módja:

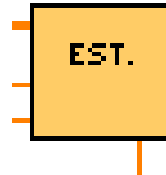
$$J_i = J_{i-1} + \Delta J_i \quad (15)$$

Az A, B, C ismeretében kiszámítható az amplitúdó, az ofszet, és a fázis. A és B négyzetes összegének a gyöke adja az amplitúdót, LSB-ben skálázva:

$$R = \sqrt{A^2 + B^2} \quad (16)$$

A C önmagában a szinusz jel egyenkomponense, kódban megadva, nem kerekítve. A fázist pedig A, B kétbemenetű arkusz-tangensének ellentettjével lehet megadni:

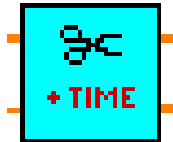
$$\varphi = -\operatorname{atan2}(B, A) \quad (17)$$



17. ábra. paraméteres becselő "sub-vi"-a

Az ábrán látható az egység, mely tartalmazza a 4 paraméteres becslési eljárást. A bemenetén megkapja a bemeneti fájlban tartalmazó adatokat, mely átmegy a túlvezérlést kezelő egységen. Továbbá az FFT eljárásból számolt periódusok számát, végül a túlvezérlés kezeléséért felelős egység által létrehozott időhalmazt. A kimenete pedig az amplitúdó, ofszet, fázis és a pontosított periódusszám. Az egység tehát meghatározza az A, B, C és J paramétereket iteratív módon, ciklusok segítségével, utána ezekből az értékekből számítja ki a paraméterek becsült értékeit.

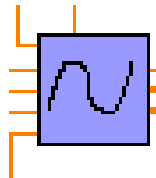
A fentebb bemutatott paraméterbecslő bemeneteit többek között a túlvezérlés okozta hibák javításáért felelős egység kimenetéről érkező adatok alkotják. Ahogy a korábbi fejezetben szó volt róla, a túlvezérelt pontokat nem szeretnénk a becselőbe eljuttatni, így szükség van valamire, mely eltávolítja ezeket a pontokat, feljegyzi a sorszámukat is. Ez nagyon egyszerűen úgy működik, hogy a program megvizsgálja a mintát, és ha az értéke nem a legkisebb, vagy a legnagyobb kimeneti kód (ezekben az esetekben lehet túlvezérlés), akkor a LabView feltételes csatornáját alkalmazva továbbengedi a program, és ezeket az értékeket egy tömbbe visszaírja. Ezzel párhuzamosan hozza létre az időtömböt, mely a megmaradt elemek sorszámát tartalmazza.



18. ábra. Túlvezérlés kezelő "sub-vi"-a

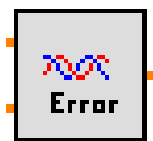
Az ábra mutatja a „sub-vi” ikonját. A bemenetére érkezik a beolvasott fájlban lévő adathalmaz, és a felhasználó által megadott bitszám. A kimenet a megmaradt minták tömbje, illetve a megmaradt minták sorszámait tartalmazó tömb.

A program, miután meghatározta a paramétereket, generál egy szinuszjelet belőle. Az eredeti jel és a generált szinuszjel különbségét pedig a felhasználó megtekintheti a kijelzőn.



19. ábra. Szinusz jel generálása becsült paraméterekből "sub-vi"

Az ábrán látható ikonú modul számítja a becsült paraméterek alapján az eredeti szinuszjelet. A bemenetről érkező amplitúdóból, ofszetből, fázisból, periódusok számából, a bitszámból és a minták számából a beépített trigonometrikus építőelemek segítségével hozza létre a mintasorozatot, melyet a kimenetre továbbít.

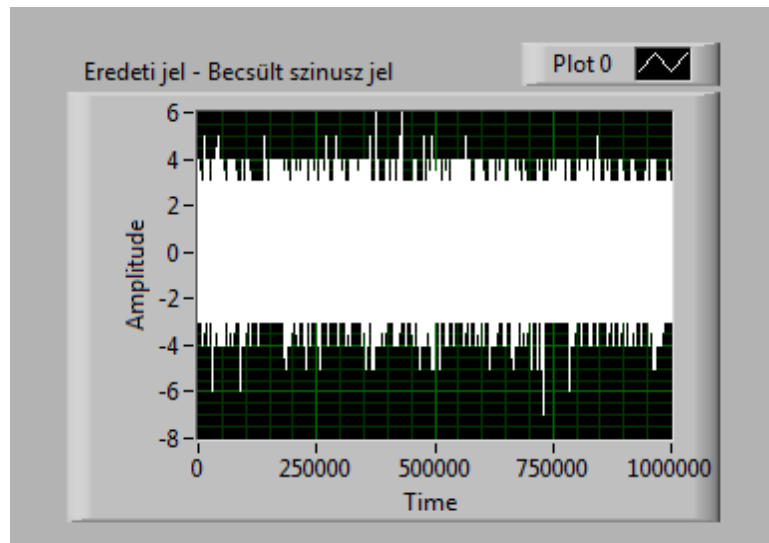


20. ábra. Becslő hiba számító "sub-vi"

Egy külön egység végzi el az egyik leglátványosabb folyamatot, a kivonást. A doboz bemenetére érkezik az előző „sub-vi” eredménye, másikkra pedig az eredeti mintasorozat, és elvégzi az alábbi feladatot:

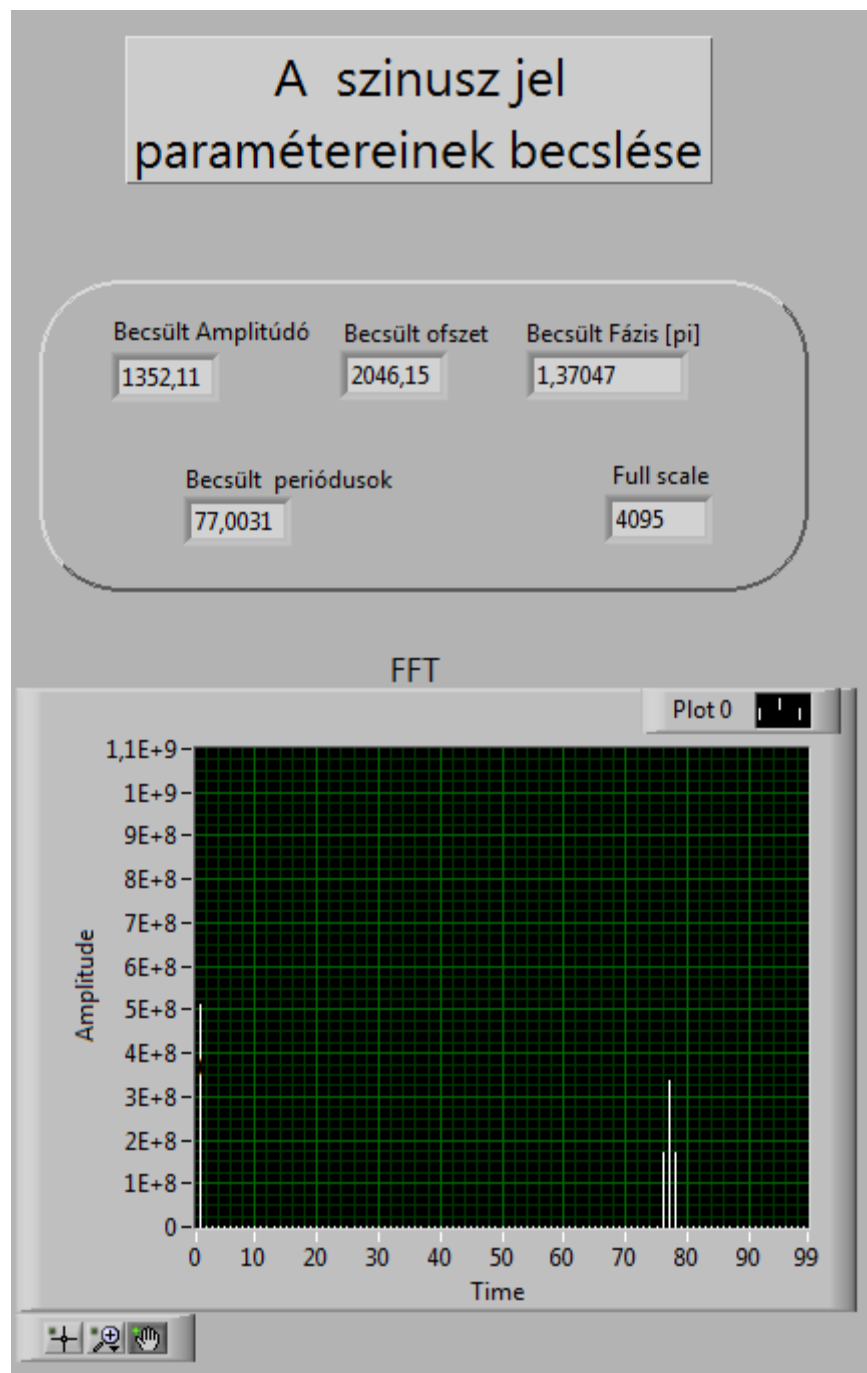
$$X[i] = X_{eredeti}[i] - X_{becsült}[i] \quad (18)$$
$$i = [0; N]$$

Itt N a minták száma.



21. ábra. Becslő hibájának kirajzolása a programban

A felhasználó természetesen a felületen leolvashatja a becsült értékeket, kódban, kvantált egységben megadva, a fázist π -ben. A program kirajzolja a FFT eredményét is, jól használható a beolvasott jel minőségének ellenőrzésére, vagy akár a periódusszám becsülésének ellenőrzésére.



22. ábra. Becslő eredményeinek, FFT eredményének kirajzolása a felhasználói felületen

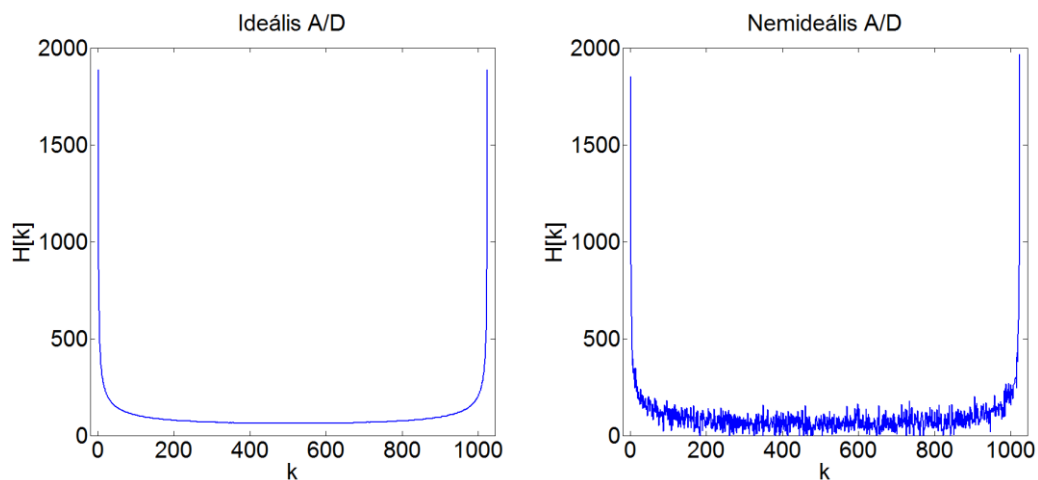
4.3 A hisztogramteszt

A komparálási szintek meghatározásához a szinuszjel gerjesztésű hisztogramtesztet választottam. Az eljárás feltételezi a szinusz paramétereinek az ismeretét. A későbbiekben majd ezért használunk szinuszillesztést. A hisztogramteszt

folyamata magába foglalja a hisztogram készítését, utána pedig a kumulatív hisztogram meghatározását.

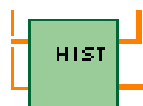
A hisztogram a mi esetünkben azt mondja meg, hogy melyik kódból hány darab található a mintasorozatban (tulajdonképpen a kódok sűrűségét mutatja meg). A hisztogramot tudjuk grafikusán is ábrázolni. Ideális szinusz mintasor hisztogramja sima kád jellegű, míg a nem ideális esetben (rosszabb jel-zaj viszony esetén) bizonytalanabb lesz a vonal. A hisztogramot a további számítások, és a megvalósítás kedvéért normáljuk a hisztogram elemeinek összegével. A normált hisztogram kiszámítási képlete, ahol $H[i]$ a hisztogram i . eleme:

$$H_{norm}[i] = \frac{H[i]}{\sum_{k=0}^{2^b-1} H[k]} \quad (19)$$



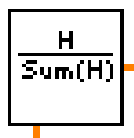
23. ábra. Ideális, nem ideális A/D esetén képzett szinusz jel hisztogram [6]

A „sub-vi” mely létrehozza a jel hisztogramját, bemenetként magát a jel adattömbjét, és a fullscale nagyságát kéri kódban. A kimeneten két tömböt szolgáltat az egység, az egyik a $H[i]$ értékei, másik pedig az ezekhez az értékekhez tartozó kódok halmaza. A LabView tartalmaz beépített sűrűségfüggvényt képező beépített „vi”-t, mely bonyolult, de igen gyors, hatékony algoritmussal készíti el a hisztogramot. Ennek a teljesítménye jóval felülmúlja a triviális „két egymásba ágyazott „for” cikluson alapuló megoldását.



24. ábra. Hisztogram létrehozó "sub-vi"

A fejlesztésben időben később felmerülő ötlet volt a hisztogram értékeinek normálása a minták darabszámára, így ez a funkció külön „sub-vi”-ban kapott helyet. A bemenetére érkező $H[i]$ értékeit egy ciklusban elosztja a minták darabszámával, aztán a kimenetre juttatja tömb formájában.

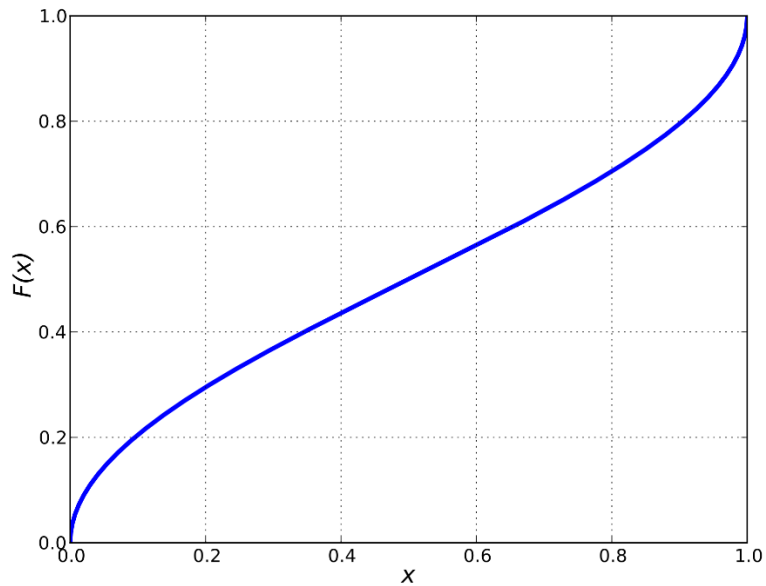


25. ábra. Normált hisztogram "sub-vi"

Ezután következik a kumulatív hisztogram számolása a hisztogramból. Ennek lényege, hogy egy adott kódhoz tartozó kumulatív hisztogram értéket úgy kapjuk meg, hogy összeadjuk a nála kisebb kódokhoz tartozó hisztogram értékeit. A kiszámítási módja:

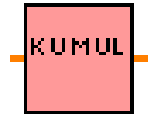
$$H_c[j] = \sum_{i=0}^j H_{norm}[i] \quad (20)$$

A kumulatív hisztogramot is tudjuk ábrázolni. Ha a normált hisztogramból képezzük a kumulatív hisztogramot, akkor egy olyan függvényt kapunk, aminek a maximális értéke 1 (lásd 26. ábra).



26. ábra Normált kumulatív hisztogram

A kumulatív hisztogramot előállító részegység bemenetére már a normált hisztogram értékei kerülnek továbbításra. A kimenet előállításához sajnos nem sikerült elkerülni az egymásba ágyazott „for” ciklusok használatát, így a futási idő számottevőbb, mint a hisztogram előállításának esetében.



27. ábra. Kumulált hisztogram "sub-vi"

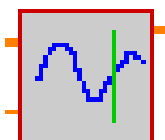
A hisztogram meghatározása előtt célszerű kezelni minta halmazunkat, amennyiben nem egész számú periódust tartalmaz a mérési fájl. Az elméleti eljárásokkal foglalkozó fejezetben volt már szó ennek a problémának megoldásáról. Ez a „sub-vi” a paraméterbecslő által meghatározott periódusszám egészrészére csonkítja a mintasorozatot a következő módon: legyen J_1 a paraméterbecslő által meghatározott periódus szám, J_2 pedig az előzőnek az egészrésze, N_1 az eredeti mintasorozat darabszáma, N_2 pedig a csonkított sorozat darabszáma. Az egy minta által lefedett fázis az alábbi formában számítható:

$$d = \frac{J_1}{N_1} \quad (21)$$

Ennek segítségével már számítható a végleges mintaszám:

$$N_2 = \text{round}\left(\frac{J_2}{d}\right) \quad (22)$$

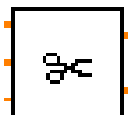
Ezen alacsony bonyolultsági szintű műveletek mellett a LabView tartalmaz beépített funkciót, mely adott intervallumot képes kiválasztani egy tömbből egységnyi műveletként, így optimális futási idő alatt fut le ez a „sub-vi”. Bemenetére a jel mintáinak tömbjét és a becsült periódusszámot várja. Kimenetét pedig a fentebb leírt eljárás által generált csonkított mintasorozat tömb alkotja.



28. ábra. Nem egész periódus levágó "sub-vi"

4.4 Az átalakító karakterisztikájának meghatározása

A kumulatív hisztogramból meghatározhatóak a komparálási szintek. Mielőtt meghatároznám őket, eldobok bizonyos értékeket. Célszerűen azokat az elemeket szoktuk eldobni, melyek 0.1 alatt vagy 0.9 felett vannak. A programot úgy valósítottam meg, hogy beállítható legyen ez a mérték. Itt van jelentősége a normálásnak, így egyszerűen csak összehasonlítjuk az elem értékét a beállított paraméterrel. Az elemek elhagyásával sajnos információt veszünk, viszont eltüntetjük a hisztogram 2 legkritikusabb pontját, a két véget. Mivel, ha például túl van vezérelve az átalakító a szinusz jellel, akkor a hisztogram szélei torzulnak el, a legkisebb, és/vagy a legnagyobb kódhoz tartozó hisztogram érték jelentősen megnő.



29. ábra Kumulált hisztogram csonkító "sub-vi"

Ez a modul LabViewban egyszerűen implementálható volt, mivel a környezet rendelkezik beépített kereső műveletekkel, melyekkel visszakereshetőek megadott értékű

elemek. Az így létrehozott egység a bemenetén kapja a kumulatív hisztogram és a hisztogram értékeihez tartozó indexek tömbjeit, továbbá a felhasználó által megadott „Cut off ratio”-t mely megadja, hogy a kumulált hisztogram széleinek milyen szélességű részét ne vegye figyelembe a program a karakterisztika meghatározásánál.

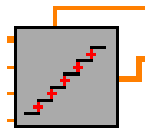
A komparálási szintek most már meghatározhatóak a kumulatív hisztogram értékeinek segítségével [2].

$$\hat{T}_k = C - R * \cos\left(\frac{\pi * H_c[k - 1]}{N}\right) \quad (23)$$

Ahol:

- **C**: a szinusz jel egyen komponense LSB-ben
- **R**: a szinusz jel amplitúdója LSB-ben
- **N**: a minták száma
- **$H_c[k]$** : a kumulatív hisztogram k-adik értéke

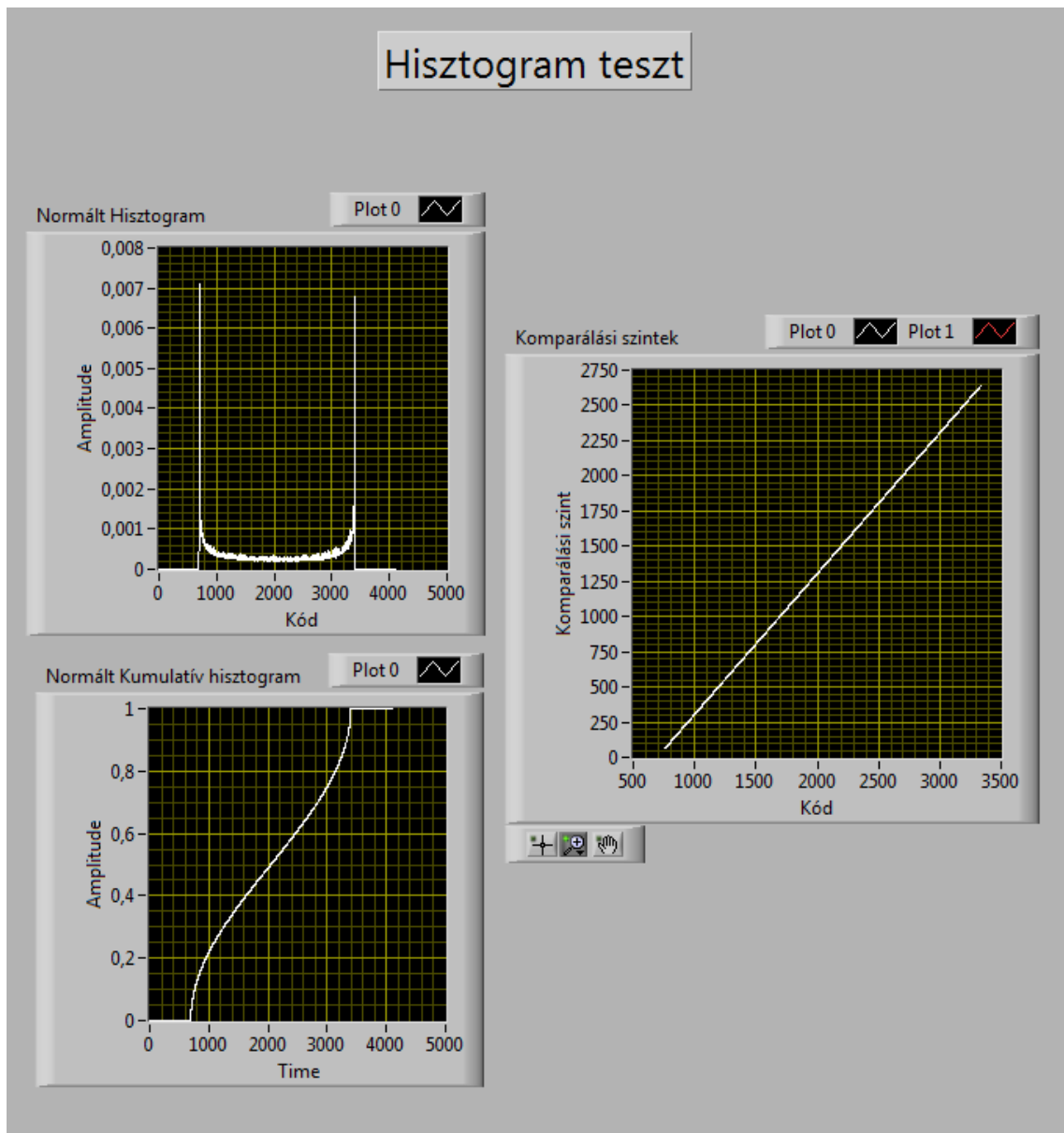
Láthatjuk, hogy a komparálási szintek meghatározásához szükségünk van a szinusz jel paramétereire.



30. ábra. Karakterisztika meghatározó "sub-vi"

Az ábrán látható a karakterisztika meghatározására szolgáló „sub-vi”. Bemenetére érkeznek a már megcsontított kumulált hisztogram adatai, a minták darabszáma, és a paraméterbecslő eredményei.

A felhasználói felületen a program kijelzi a hisztogram és a kumulált hisztogram értékeit, a meglévő indexek függvényében. Az előzőleg kifejtett, nem koherens mintavételezést kezelő eljárás úgy lett megvalósítva, hogy a felhasználó a hisztogramok és a döntési szintek kirajzolását az eredeti mintasorral, illetve a koherensé tett sorozattal meg tudja tekinteni. Így prezentálja a program a koherens mintavételezés fontosságát.

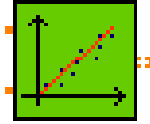


31. ábra. Hisztogramteszt a felhasználói felületen

4.5 Az átalakító hibáinak számítása

A komparálási szinteket ismervén tudunk következtetni az átalakító tulajdonságaira, a linearitási és nemlinearitás hibák számszerűen jellemezhetőek lesznek. Ezen nemlinearitási hibákat egy referencia egyenes segítségével nyerem. A referencia egyenes képzésének is több módja lehet. Szóba jöhet a végpontokra illesztett egyenes, de ez a módszer elég pontatlan, és ha extrémebb értékűek lennének a komparálási szintek végei, akkor nagy hibát eredményezne. Ezért egy úgynevezett „least squares” egyenes illesztést választottam. Az alábbi módon számolható ki a referencia egyenes meredeksége, és ofszetje:

$$\begin{bmatrix} \text{Merekség} \\ \text{Ofszet} \end{bmatrix} = \begin{bmatrix} 1 & 1 \\ 2 & 1 \\ \vdots & \vdots \\ 2^{b-1} & 1 \end{bmatrix}^{\dagger} * \begin{bmatrix} T_0 \\ T_1 \\ \vdots \\ T_{2^{b-1}} \end{bmatrix} \quad (24)$$



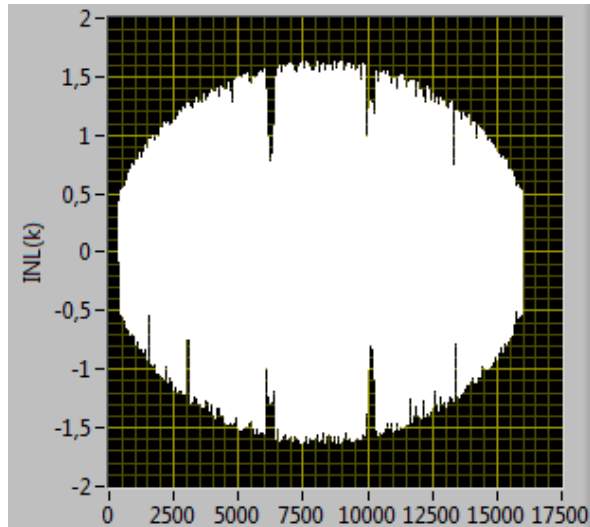
32. ábra. Egyenes illesztő "sub-vi"

Ezen feladatokat elvégző modul a karakterisztika értékeit és a kumulált hisztogram csonkítását végző egység által létrehozott indexeket tartalmazó tömböket várja bemenetként. A LabView sokrétű mátrix művelet apparátussal rendelkezik, így a fentebb leírt eljárás könnyen implementálható volt beépített „sub-vi”-k segítségével, kielégítő futási időt elérve. Kimenetként az illesztett egyenes merekségét és ofszetjét szolgáltatja.

A referencia egyenes felhasználásával meghatározhatjuk a nemlinearitási hibákat. Az integrális nemlinearitási hibát számítom ki először, mert ennek az eredményeit fel lehet használni a későbbiekben. Az integrális nemlinearitás megadja a referencia egyenes és a valódi karakterisztika pontjainak távolságát. Egy b bitszámú átalakító esetén $2^b - 1$ darab távolságot számolunk, ezek a távolságok közül általában a leghosszabbra vagyunk kíváncsiak, ez jellemzi az átalakító *worst case* hibáját. Továbbá LSB-re normálva a legbeszédesebb.

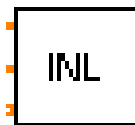
$$INL[k]_{LSB} = \frac{y[k] - T[k]}{LSB} \quad (25)$$

Itt $y[k]$ a referencia egyenes k -edik pontja, $T[k]$ a komparálási szintek k -edik eleme.



33. ábra. Szimulált ideális A/D INL diagram

Az ábrán egy 14 bites szimulált ideális A/D átalakító INL grafikonját látjuk. Észrevehető, hogy a hiba a középső kódoknál a legnagyobb. Ez azért van, mert a szinusz jel meredeksége ott a maximális, tehát a legkevesebb minta oda fog kerülni a hisztogramban, ezáltal pontatlanabbak ott az információink.



34. ábra. INL számító "sub-vi"

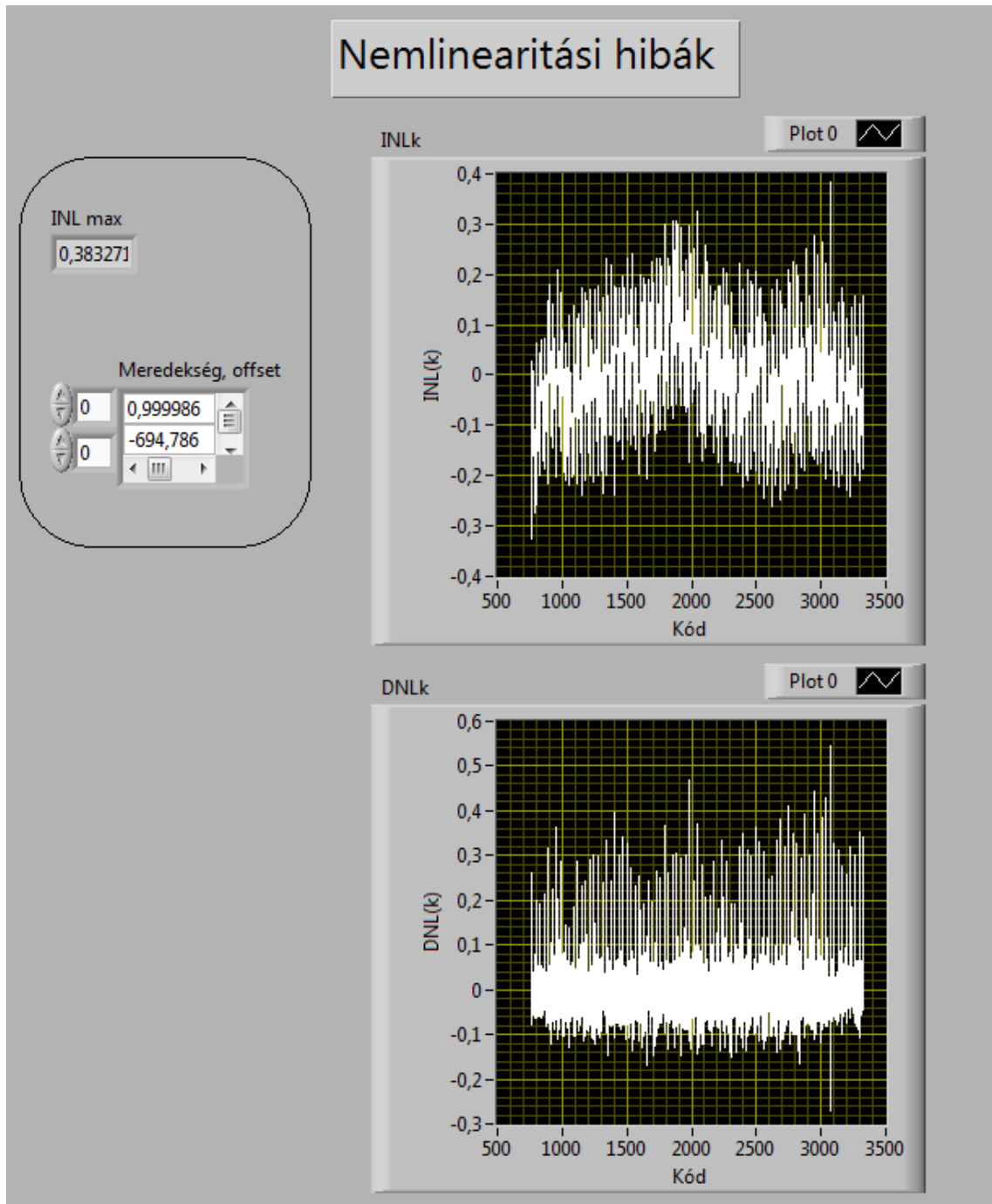
A megvalósított „sub-vi” bemenete a komparálási szintekből álló tömb mellett az illesztett egyenes paramétereit, a „cut off” egység által csonkított kódok halmazát várja a bemenetre. A kimeneteket pedig az INL értékek, és a maximális INL érték alkotja.

A differenciális nemlinearitás azt mondja meg, hogy a karakterisztika egy lépcsőjének a kódszélessége mennyire tér el az LSB-től. Ezen eltérések közül is általában a maximálisra vagyunk kíváncsiak. A DNL értékek számíthatók a már ismert INL értékekből is.

$$DNL[k]_{LSB} = \frac{INL[k + 1] - INL[k]}{LSB} \quad (26)$$



35. ábra. DNL számító "sub-vi"



36. ábra. INL, DNL megjelenítése a felhasználói felületen

5 Értékelés, saját mérések eredményei

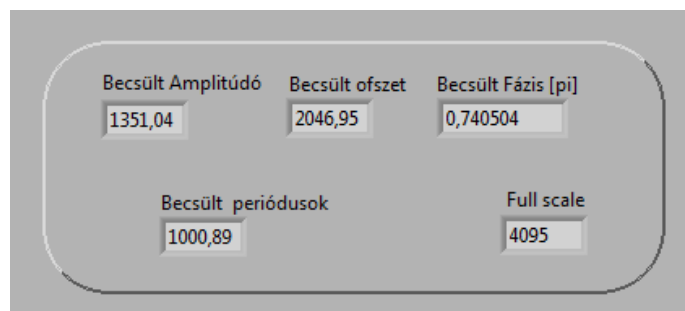
1. Mérés

A névleges paraméterek:

- Bitszám: 12
- PkPk/FS: 0.7
- Offset/FS: 0
- Periódusszám: 1001
- Mintaszám: 1000000
- Cut-off ratio: 0.1

Becslés eredményei:

A paraméterek:



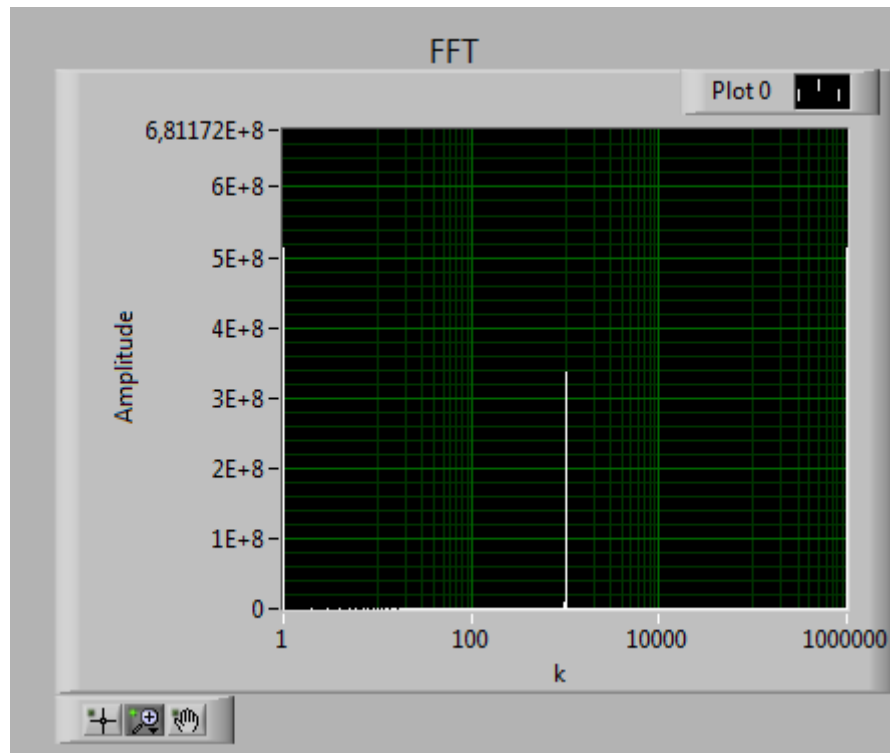
37. ábra. Paraméter eredmények

Az eredmények helytállnak. Mivel az ofszet névlegesen FS/2, az amplitúdó pedig:

$$0,7 * \frac{(2^{12}-1)}{2} = 1433$$

A becslt amplitúdó alapján a PkPk/FS arány körülbelül 0,66 csak, nem annyi, mint a névleges. A becslés hibájánál majd látjuk, hogy nem a becsló hibázott, hanem a névleges adatok álltak rosszul rendelkezésre.

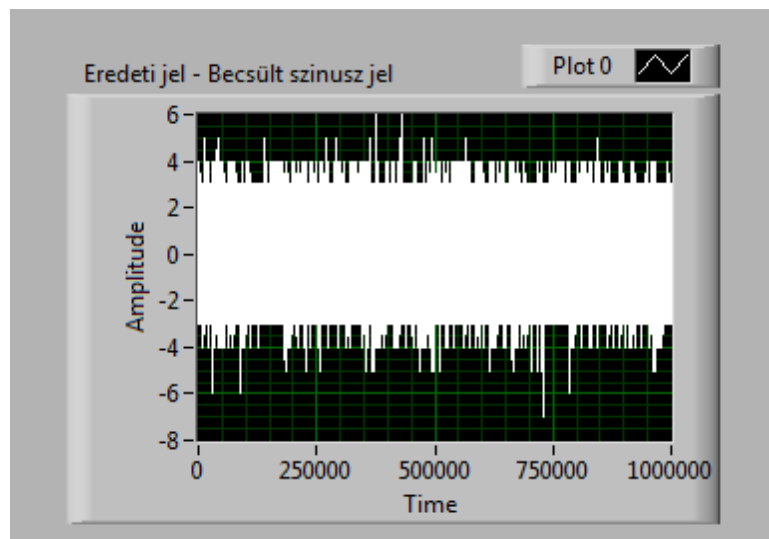
FFT:



38. ábra. FFT eredménye

FFT eredményéből látható, hogy viszonylag tiszta szinuszjellel van dolgunk, tisztán leolvasható a periódusok száma.

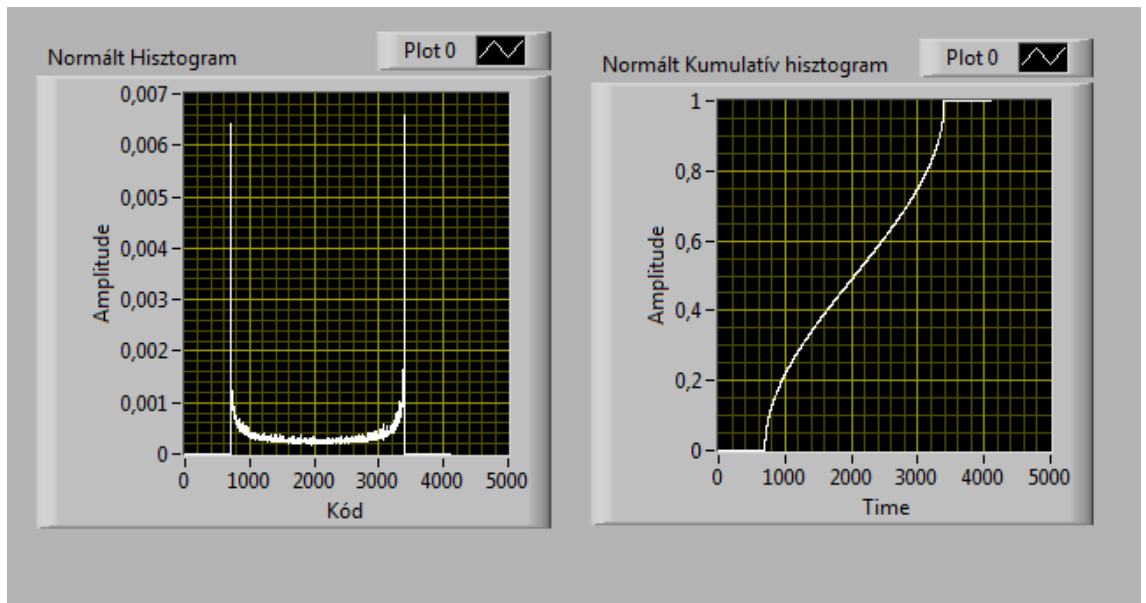
Becslés hibája:



39. ábra. Becslés hibája kirajzolva

A becslés hibája amplitúdóhoz viszonyítva: $\frac{6}{1351} = 0,444 \%_{ampl.}$

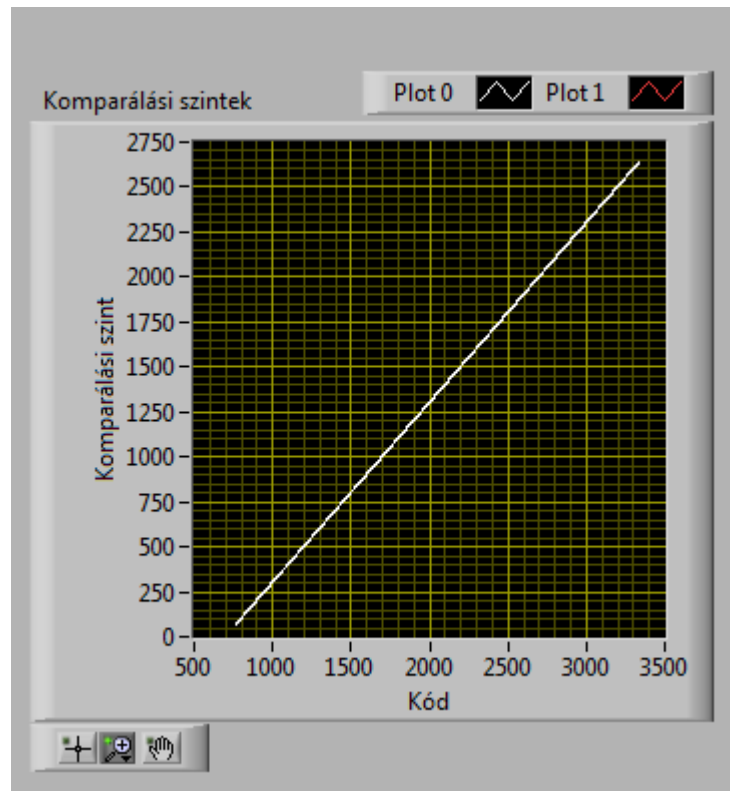
Hisztogramok:



40. ábra. Hisztogramteszt

A hisztogramok kezdeti értéke az ofszet mértékében el van tolódva, látszik hogy nem volt teljesen gerjesztve az A/D.

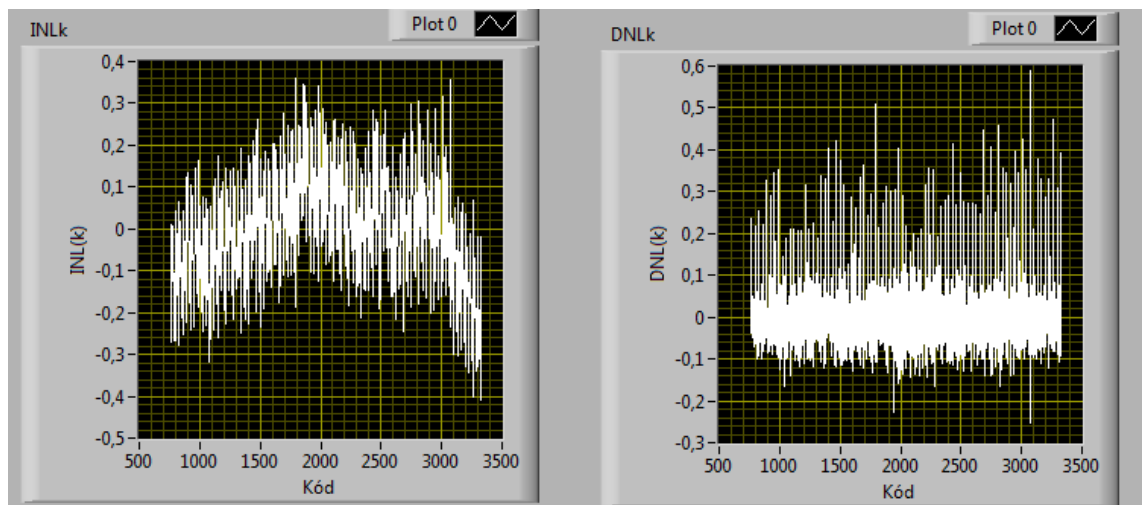
A karakterisztika:



41. ábra. Komparálási szintek

Szép egyenes karakterisztikát kaptunk.

INL,DNL:



42. ábra INL-DNL

Kis mértékű nemlinearitási hibákat tapasztaltunk.

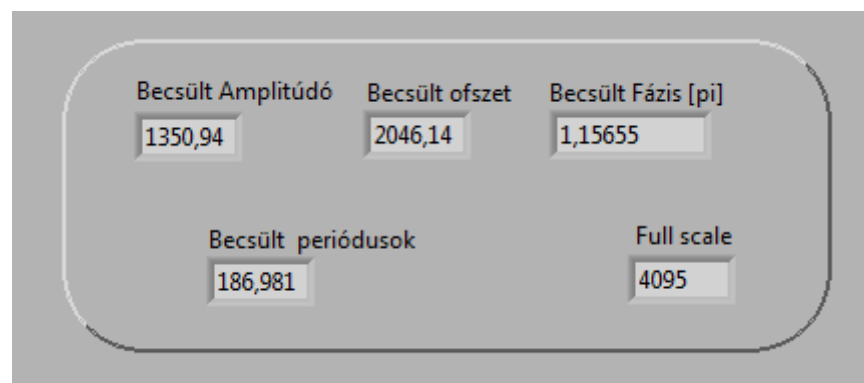
2. Mérés

A paraméterek:

- Bitszám: 12
- PkPk/FS: 0.7
- Offset/FS: 0
- Periódusszám: 187
- Mintaszám: 1000000
- Cut-off ratio: 0.05

Becslés eredményei:

A paraméterek:



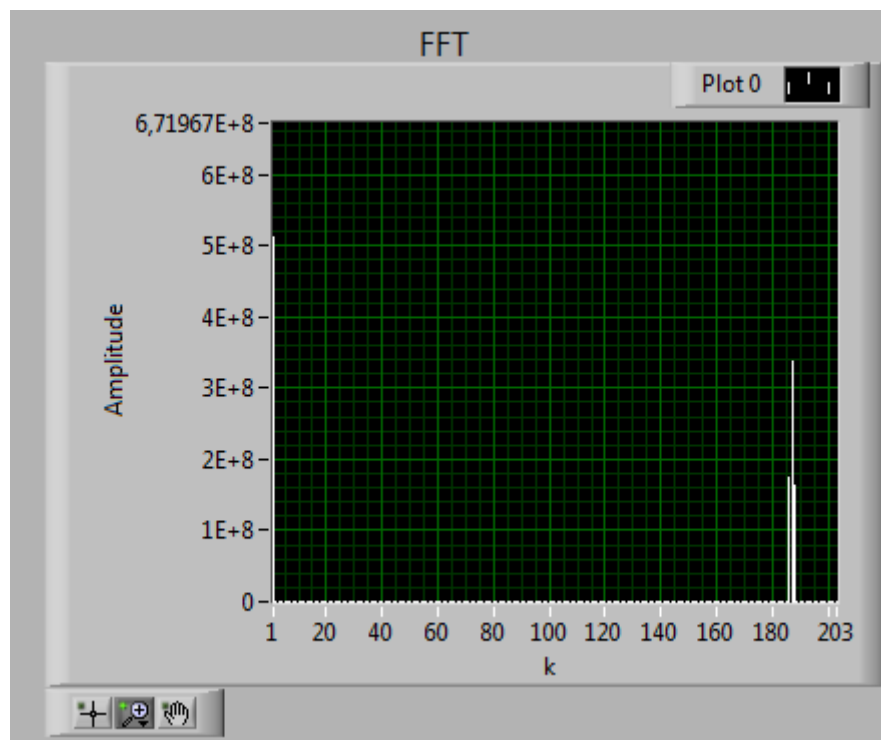
43. ábra. Becsült paraméterek eredményei

Az eredmények ugyanúgy helytállnak, mint az első mérésnél. Mivel az ofszet névlegesen FS/2, az amplitúdó pedig:

$$0,7 * \frac{(2^{12}-1)}{2} = 1433$$

A becsült amplitúdó alapján a PkPk/FS arány körülbelül 0,66 csak, nem annyi, mint a névleges. Itt ugyancsak az a helyzet, mint az előző mérés esetén, hogy nem a becslő hibázott, hanem a névleges adatok álltak rosszul rendelkezésre.

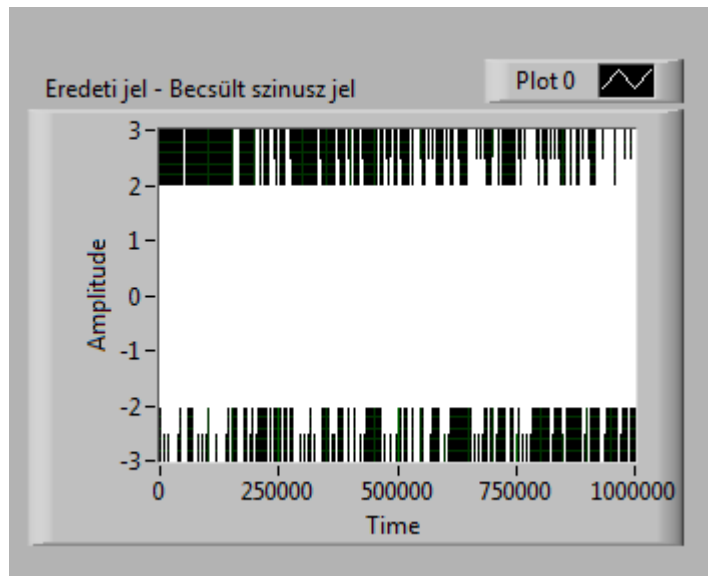
FFT:



44. ábra. FFT eredménye

Az FFT eredményéből látható, hogy viszonylag tiszta szinuszzel van dolgunk, tisztán leolvasható a periódusok száma.

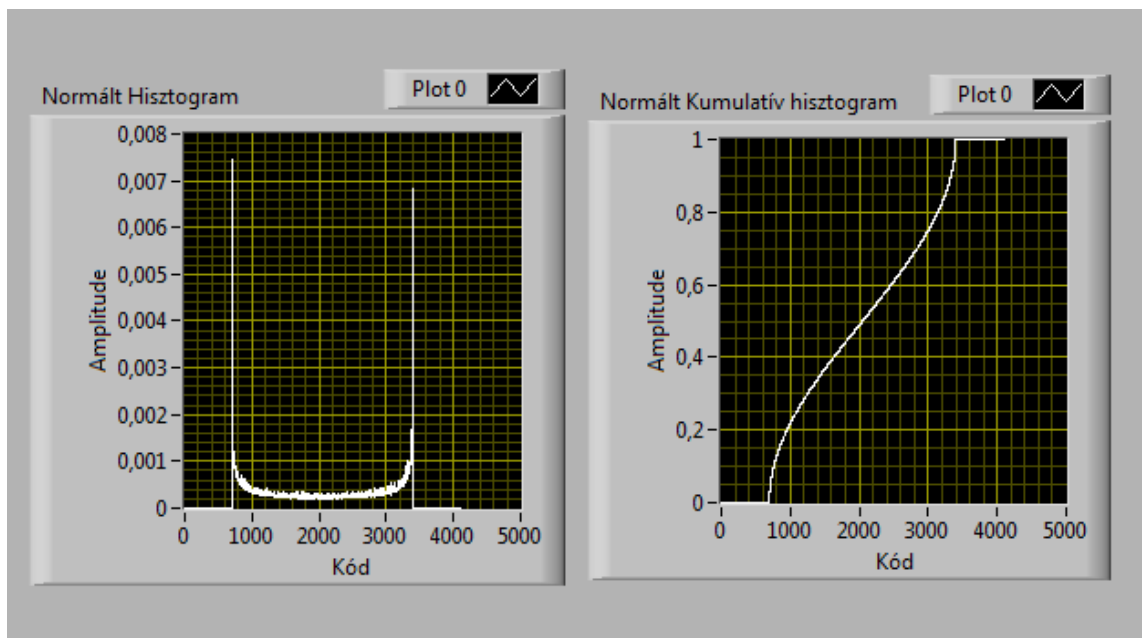
Becslés hibája:



45. ábra. Becslés hibája kirajzolva

A becslés hibája amplitúdóhoz viszonyítva: $\frac{3}{1351} = 0,222 \%_{ampl}$

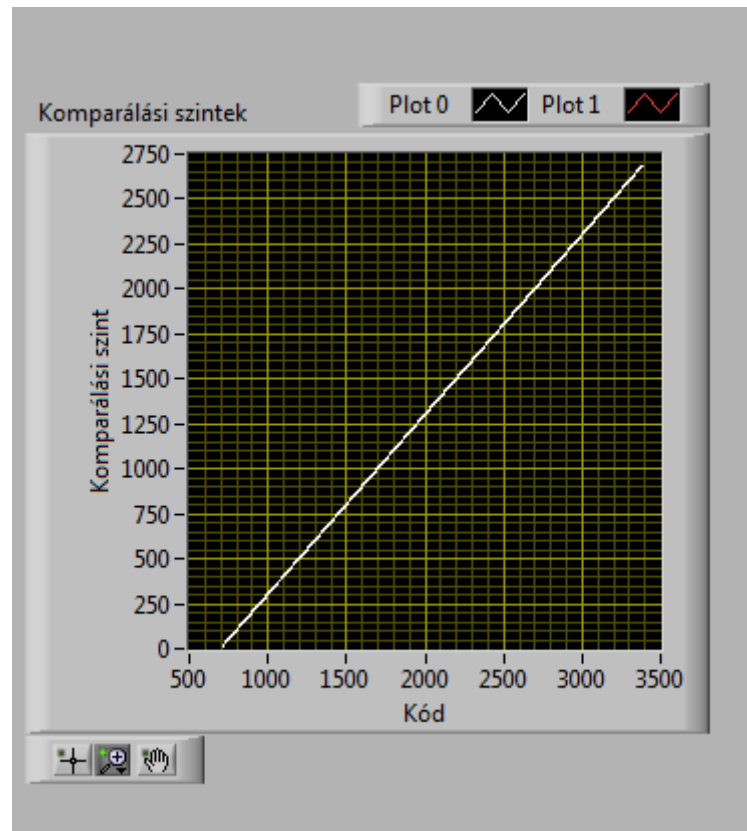
Hisztogramok:



46. ábra. Hisztogramteszt

A hisztogramok kezdeti értéke az ofset mértékében el van tolódva, továbbá látszik, hogy nem volt gerjesztve az A/D a teljes tartományon.

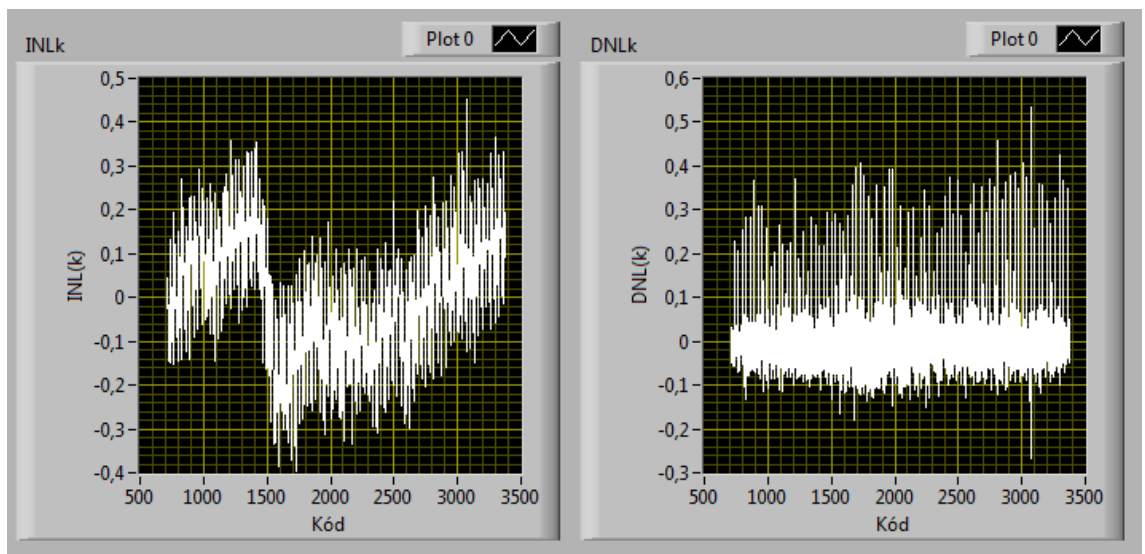
A karakterisztika:



47. ábra. komparálási szintek

A karakterisztika egyenes, pont amire számítottunk.

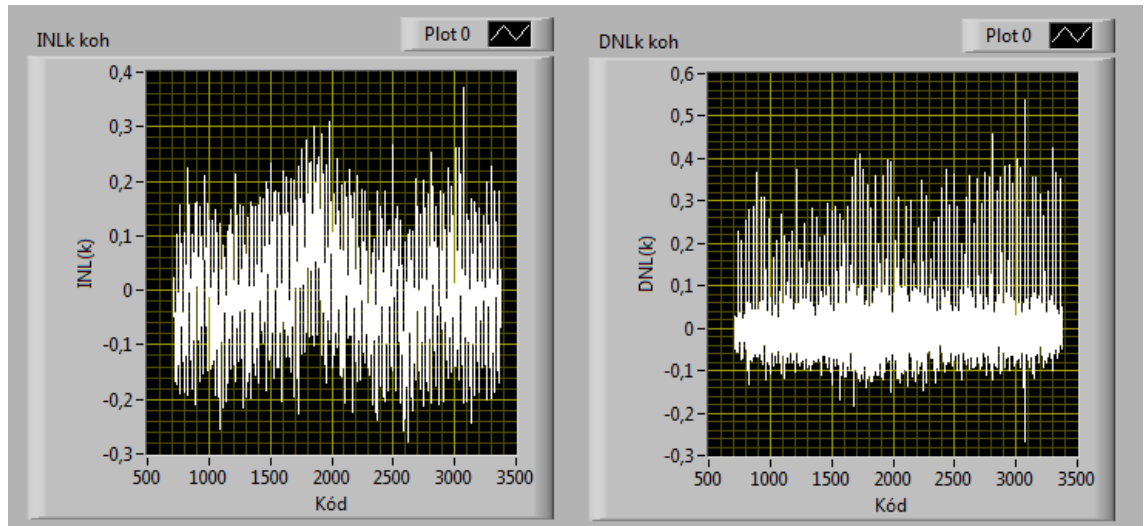
INL,DNL:



48. ábra. INL-DNL

Kis mértékű nemlinearitási hibákat tapasztaltunk, viszont a koherenssé alakító programrészeken való átfuttatás után még jobb eredményeket tapasztaltunk.

Koherensé téve:



49. ábra. INL-DNL koherens esetben

0.05-el alacsonyabb maximális INL-t kaptunk eredménynek

6 Összefoglalás

6.1 Az elkészült feladat értékelése, a fejlesztés lépései

A projekt elkezdésekor még nem voltak LabView ismereteim. Így az első dolgom a fejlesztői környezet megismerése volt. Az első hetek kutatása ebben a témában merült ki, eleinte az interneten fellelhető „tutorial”-ok segítségével. Ezután az első lépés a munkámban a hisztogramteszt megismerése volt. Az elmélet áttanulmányozása után, implementáltam a hisztogram, és a sűrűségfüggvény képzését. Ezt, hogy tesztelni tudjam, létrehoztam egy A/D átalakítót szimuláló progamegységet, mely beállítható paraméterekkel rendelkező kvantált szinuszjelet adott a kimenetén.

Amint a hisztogram működőképes lett, a kumulált hisztogram, és a karakterisztika meghatározásának megvalósítása volt a feladat. Ezek nem okoztak gondot, mivel alapvető matematikai műveletekkel megoldhatóak. Az egyenes illesztésénél viszont az „LS” becslést választottam, melynek áthelyezése a programba már nem volt teljesen triviális feladat. Viszont a LabView környezet sokrétű mátrix műveleteket tartalmaz, amikkel meg tudtam oldani a problémát. Az INL, DNL hibák meghatározása ugyancsak nem jelentettek gondot.

A paraméterek becslési módszereinek elméleti áttanulmányozása után, neki láttam a periódusok számának meghatározásának. Nem volt könnyű dolgom, többször is elakadtam, végül a beépített FFT funkció használatával sikerült a feladat. Ezután eleinte a 3 paraméteres becslési eljárást implementáltam, ez nem jelentett problémát. A program készen állt, hogy egy szimulált A/D mintáiból becsüljön, majd a becsült paraméterekből hibát számoljon. Több apró csiszolás, mint például a „cut-off” funkció beiktatása, illetve a hisztogramteszt javítása után a várt eredményeket hozta az alkalmazás.

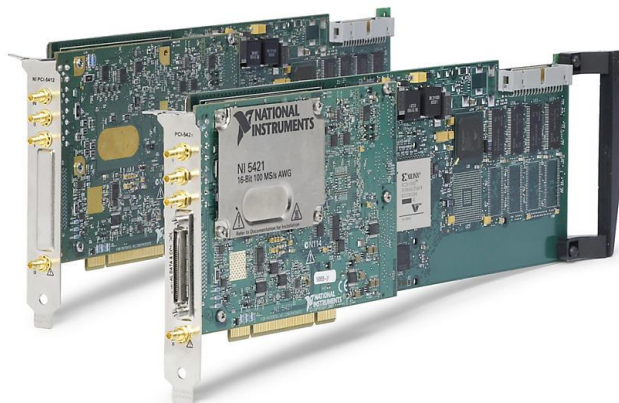
A program ekkor még nem tudta kezelni megfelelően a túlvezérlést, és a nem koherens mintavételezést, továbbá a már ismert 4 paraméteres becselő sem volt megvalósítva. Így ezeknek a feladatoknak álltam neki a program fejlesztése érdekében. Első lépésként a négy paraméteres becslési eljárást építettem bele a programba a három paraméteres eljárást lecserélve. Ez nem volt triviális, több konzultálás, átgondolás után sikerült csak, de pontosabb eredményeket adott, mint az elődje. Következő lépésként a túlvezérlés kezelését tűztam ki célul. A kivitelezés nem jelentett akadályt, a program ezután könnyedén kezelte a „full-scale”-nél nagyobb tartományt kihasználó

szinuszjeleket. Hasonlóan a nem egész számú periódusokat kezelő egység megvalósítása sem jelentett gondot.

Ekkor tehát a program szinte a mai verziójában létezett. Azonban nem volt képes valóságos átalakítóról érkező adatokat beolvasni, így a szimulált átalakítót lecseréltem egy fájl beolvasó modulra. A program tehát képes lett elvégezni a specifikációban leírt követelményeket, teljesítve a feladatát.

6.2 Továbbfejlesztési lehetőségek

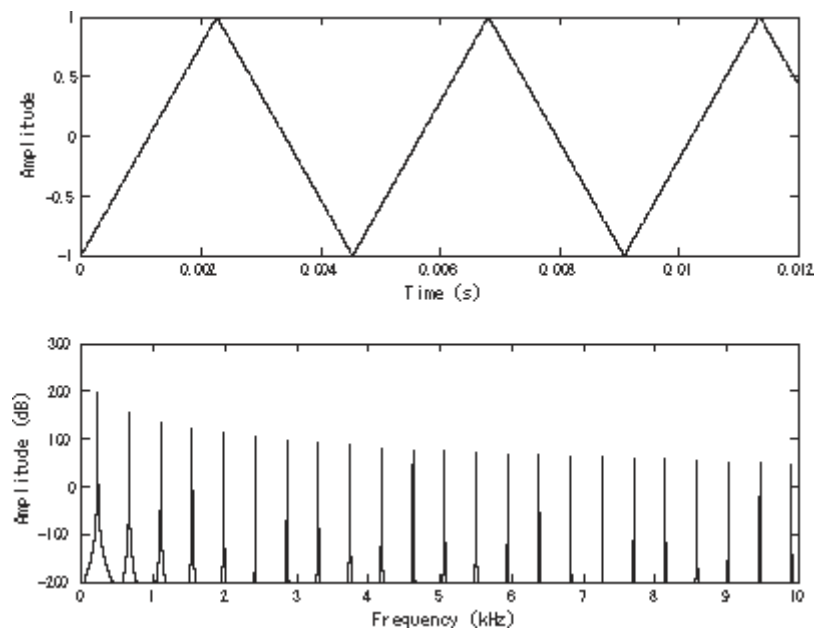
A program a jelenlegi állapotában „offline” tesztelést végez el. Tehát, egy generátorral létrehozunk egy jelet, átfuttatjuk egy átalakítón, majd a minták eredményeit fájlba mentjük, majd ezután olvassuk be a programba az értékeket. Ez a folyamat kissé körülményes is lehet, emiatt felmerülhet a kérdés, hogy felhasználóbarátibbá, gyorsabbá tegyük ezt a folyamatot. Megfelelő eszközök segítségével a programot át lehetne alakítani „online” teszteléshez. A program a gerjesztő jelet előállítaná, kiadná egy D/A-ra, majd valós időben figyelné a bejövő adatokat, melyeken végezné a tesztelési eljárást. Így az adatok tárolására, továbbítására nem lenne gondunk. Azonban ahhoz, hogy ezt meg lehessen valósítani, szükségünk lenne egy kiegészítő hardverre, egy jelgenerátorra, melyet a *National Instruments* forgalmaz, ahogyan LabView fejlesztői környezetet is, így biztosan teljesen kompatibilis lenne a programmal. Ezáltal a méréseink elkészítése roppant módon egyszerűsödne, a számítógépünkről közvetlenül tesztelhetnénk az átalakítót.



50. ábra. Számítógépbe építhető jelgenerátor

A tesztelés futási ideje nagyobb bitszámok illetve nagyobb számítási kapacitással rendelkező számítógép esetén is zavaróan nagy lehet. Ezt csökkentve, a becslési eljárások algoritmusait lecserélve lehetne orvosolni. Jelenleg a becslés időtartományban zajlik. Ha olyan becslési módszert választanánk a jel paramétereinek meghatározására, mely frekvenciatartományban működik, kisebb számítás igényű programot kapnánk eredményül. Ezáltal a program is gyorsabban végezné a feladatát, felhasználóbarátibbá válna. Az előbb bemutatott valós idejű teszteléshez minden bizonnyal alkalmazni kellene ezt a módosítást is, hiszen abban az esetben kritikus szempont lehet a program futási ideje.

Az elméleti háttérnek szentelt fejezetben ki lett fejtve korábban, hogy miért szinuszos gerjesztést alkalmaztam. A program viszont kiterjeszhető lenne más jelek kezelésére is, például háromszögjel, vagy gauss zaj. Ehhez a hisztogramteszt folyamatán kellene változtatni, módosítani. Ezekben az esetekben nem csak egy kitüntetett frekvencián gerjesztenénk az átalakítót, hanem ismerve a jelek spektrális képét, szélesebb sávzélességen.



51. ábra. Háromszögjel idő, és frekvenciatartományban

7 Irodalomjegyzék

- [1] 1241-2010, I. S. (2011. január 14). IEEE Standard for Terminology and Test Methods for Analog-to-Digital Converters.
- [2] atx7006. (2015). Terms A/D converter characterization. Forrás: <http://www.atx7006.com/articles/terms/adc>
- [3] Electronics tutorials. (2016). Amplitude Distortion due to Clipping. *Amplifier Distortion*. Forrás: http://www.electronics-tutorials.ws/amplifier/amp_4.html
- [4] John, P. (2003). Basic acoustics and Signal Processing. Forrás: <http://www.linuxfocus.org/English/March2003/article271.shtml>
- [5] Ming, X., & Ding, K. (1996). Corrections for frequency, amplitude and phase in a fast Fourier transform of a harmonic signal. *Mechanical Systems and Signal Processing*.
- [6] Pálfi, V. (2015). Analóg-digitális átalakítók hatásos tesztelése a szinuszos gerjesztő jel paramétereit becsülő eljárással. *Disszertáció*. Budapest.
- [7] Walt, K. (2007. december 5). ADC and DAC Static Transfer Functions and DC Errors. *Analog Devices*. Forrás: <http://www.embedded.com/print/4015837>