

Mérési útmutató a
Beágyazott és ambiens rendszerek laboratórium (vimia350)

Elosztott rendszerek és szenzorhálózatok 1.

című méréséhez

Készítette:
Orosz György, Dr. Sujbert László
BME-MIT

2008. március

9. mérés

Elosztott rendszerek és szenzorhálózatok 1.

9.1. A mérés célja

Manapság egyre több területen találkozunk úgynevezett elosztott mérőrendszerekkel. Ezen rendszerek sajátossága, hogy amíg a hagyományos rendszerekben a mérési adatok gyűjtését egy központi egység vezérli, addig ezen esetekben a mérést a szenzorok többé-kevésbé autonóm módon hajtják végre, nincs szigorú értelemben vett központi felügyelet. Az egységek között általában nincs közvetlen kapcsolat, az adatcsere valamilyen megosztott kommunikációs csatornán történik. Ilyen kommunikációs csatorna lehet például az Ethernet vagy a manapság egyre jobban terjedő különféle rádiós kommunikációs protokollok (pl. ZigBee, Bluetooth...). A hálózati kommunikáció előnye, hogy nem szükséges közvetlen fizikai kapcsolat kiépítése minden egyes szenzor és a központi egység között (mely nagyméretű rendszer esetén igen költséges lehetne), hanem egyetlen közös csatorna használata lehetséges, és akár meglévő hálózatot is igénybe lehet venni. A vezeték nélküli kommunikáció pedig tovább növeli a flexibilitást az által, hogy a szenzorok pozíciója nem kötött, bizonyos feltételek mellett szabadon mozoghatnak.

Az eltérő konfiguráció miatt természetesen meg kell birkóznunk a decentralizáció miatt felmerülő problémákkal, mely alapvetően az egyes részegységek működésének összehangolását (szinkronizálását) jelenti.

A mérés során egy egyszerű elosztott mérőrendszerben felmerülő kérdésekkel foglalkozunk. A rendszer esetünkben egy vezeték nélküli szenzorhálózat. Az alapvető feladatok közé tartoznak a szinkronizációval kapcsolatos kérdések, illetve a mért jelek feldolgozása.

9.2. A mérőrendszer felépítése

Mivel a mérési feladatok elvégzéséhez a mérőrendszer bizonyos szintű ismerete elengedhetetlen, ezért először a rendszer működésével kapcsolatos részletek kerülnek bemutatásra.

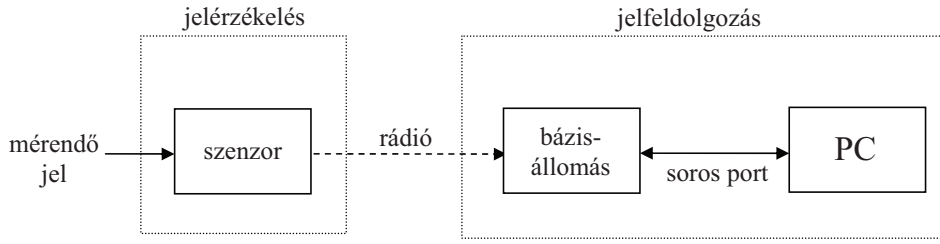
A mérés során használt mérőrendszer logikai felépítése az 1. ábrán látható. A mérőrendszer egy olyan vezeték nélküli szenzorhálózat, amely motókból épül fel. A motók a mérendő jel érzékelését egy szenzorkártya segítségével végzik, a kommunikáció pedig egy ZigBee kompatibilis rádiós kártya segítségével történik. A szenzorkártya a rajta elhelyezett mikrofonoknak köszönhetően alkalmas akusztikus jelek vizsgálatára, illetve lehetőséget nyújt bármilyen analóg jel csatlakoztatására a „Line-In” bemenete segítségével.

A mérés során vigyázzunk, hogy 1,6 V-nál jelentősen nagyobb amplitúdójú jelet ne adjunk ezen bemenetre, valamint arra, hogy a motók kikapcsolt állapotában semmilyen jel ne kerüljön a bemenetre (ilyenkor tanácsos eltávolítani a jelgenerátorhoz vezető csatlakozót).

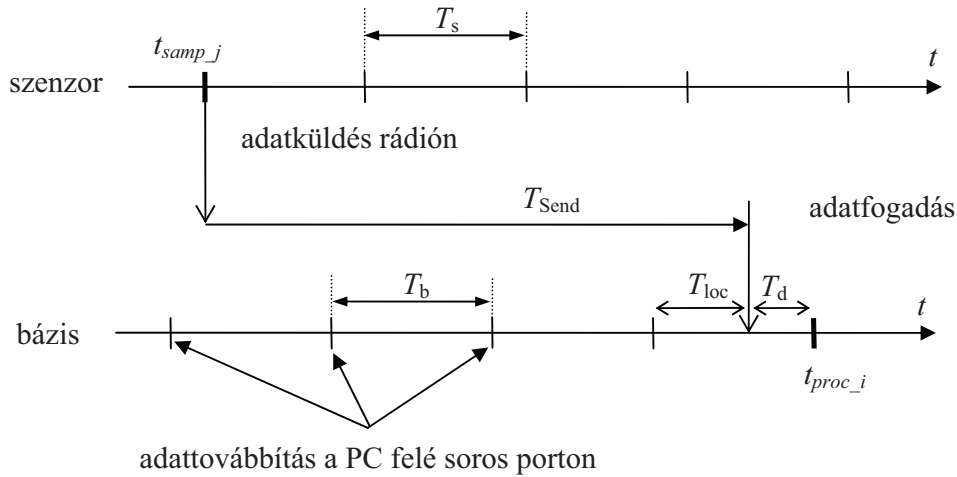
A mért jelek digitalizálása a motókon található mikrokontroller (μC) analóg-digitális átalakítójával (ADC) történik. Az összegyűjtött adatokat a bázisállomás segítségével juttatjuk el a PC-hez.

A működést a 2. ábra idődiagramja szemlélteti. A felső időtengely a szenzorhoz, az alsó pedig a bázisállomáshoz tartozik. Az időtengelyen található függőleges vonalak a szenzor esetén a mintavételi időpontokat, a bázisállomás esetén az adatok a PC felé történő továbbítását jelentik. T_{Send} az adatok rádióan keresztül történő továbbításához szükséges időintervallumot jelöli.

A rendszer működése részletesebben a következő. A szenzor folyamatosan mintavételezi a mérendő jelet $f_s = 1800$ Hz-es mintavételi frekvenciával. A mintavételi időpontok közötti távolság tehát $T_s = 1/f_s$. A mintavételezés teljesen autonóm módon történik a motón található μC harveres timer egysége által időzítve, oly módon, hogy 1800 Hz-es ütemben megszakításokat generál, és az AD konverzió (mintavételezés) a megszakítások alkalmával történik. A szenzor a mintavételezett értékeket rádióan keresztül folyamatosan továbbítja a bázisállomás felé 25 adatból álló csomagokban.



1. ábra. A mérőrendszer felépítése



2. ábra. Működési idődiagram

A szenzorhálózathoz érkező adatok begyűjtését (fúzióját) a bázisállomás végzi, melynek működése szintén periodikus: feladatát T_b időközönként (tehát $f_b = 1/T_b$ frekvenciával) hajtja végre. f_b értéke a bázismóton található kapcsolók segítségével 1800 Hz illetve 1820 Hz értékekre állítható. Az ütemezést a bázisállomás esetében is a saját timere végzi, mely T_b időközönként megszakítást generál. A megszakítások alkalmával végrehajtott feladat a szenzortól érkezett utolsó és utolsó előtti adat, valamint beállítástól függően vagy az adat érkezési idejének (T_{loc}), vagy az utolsó minta keletkezése óta eltelt idő (T_d) továbbítása soros porton a PC felé. T_{loc} illetve T_d a szinkronizációban játszik majd fontos szerepet. T_{loc} a bázisállomás utolsó megszakítási időpontjától számítva értendő (ld. 2. ábra). A PC végzi a kapott adatok fájlba mentését, melyek feldolgozása a mérés során MATLAB segítségével történik.

Mivel a PC-n végzett jelfeldolgozáshoz szükséges minden adatot a bázisállomás szolgáltatja, a PC-n megvalósított jelfeldolgozási műveletek akár a motokon is végrehajthatóak lennének megfelelő számítási kapacitás esetén. A bázisállomás-PC pár tehát úgy is felfogható, mint egy nagyteljesítményű műveletvégző egység, mely a szenzorok adatait folyamatosan fogadja és feldolgozza. A jelfeldolgozási időpontok ebben az esetben a bázisállomástól a PC felé történő adattovábbítások időpontjai, melyeket a 2. ábrán t_{proc_i} -vel jelöltünk. A későbbiekben tehát a bázisállomás adattovábbítási időpontjaira mint jelfeldolgozási pontokra fogunk hivatkozni.

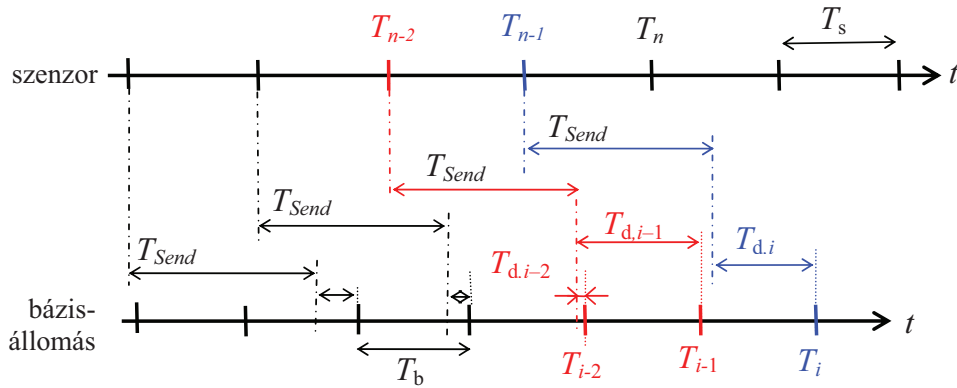
9.3. Szinkronizáció

A szinkronizáció alapvetően olyan alkalmazások esetén játszik fontos szerepet, ahol az egyes események időpontjainak meghatározása kritikus fontosságú az algoritmus végeredménye szempontjából. Jelfeldolgozási szempontból egy esemény lehet a megfigyelt jel paramétereiben bekövetkezett egy vagy több változás, például az, hogy a jel értéke eléri az 1 V feszültséget.

A mérés során két alapvető esetet vizsgálunk. Egyik esetben a jel érzékelésének és feldolgozásának névleges frekvenciája megegyezik (a konkrét rendszerben 1800 Hz), a másik esetben a jelérzékelés és a feldolgozás frekvenciája eltérő (esetünkben 1800 Hz és 1820 Hz). Mindkét esetben a problémát az okozza, hogy mivel a szenzor és a bázisállomás egymáshoz képest aszinkron működik, a szenzoroktól érkező adat a bázisállomás két jelfeldolgozási időpontja között bármikor beérkezhet. Ezt az időpontot adja meg T_{loc} értéke. Ha a jelfeldolgozási időpontokban csupán a beérkezett adatokat vennénk figyelembe, akkor a jelérzékelés és feldolgozás közötti idő állandóan változna. Ezt szemlélteti a 3. ábra, melyen a szenzor és a bázisállomás névleges frekvenciái megegyeznek ($T_s = T_b$).

A valóságban azonban $T_s = T_b$ nem biztosítható, hiszen a szenzorok működéséhez szükséges, és azok időzítési viszonyait meghatározó órajelet előállító kvarcoszcillátorok frekvenciái eltérnek egymástól. Habár ez az eltérés nem nagy, jellegzetesen néhány ppm, hosszú idő alatt jelentős hibához vezethet. A 3. ábrán a $T_b < T_s$ eset látható. Természetesen a jelenség akkor is fennáll, ha T_b és T_s között nagyobb az eltérés.

Amennyiben megvizsgáljuk a 3. ábrát, láthatjuk, hogy a T_{i-2} -vel jelölt jelfeldolgozási és a hozzá tartozó T_{n-2} -ben bekövetkező mintavételi idő között a késleltetés $T_{kesl1} = T_{Send} + T_{d,i-2}$. A T_{i-1} jelfeldolgozási időpontban viszont a késleltetés értéke $T_{kesl2} = T_{Send} + T_{d,i-1}$. A két jelút közti különbség szemmel látható, és a $(T_{d,i-2} - T_{d,i-1})$ képlettel számítható, mivel az üzenetküldés ideje (T_{Send}) állandónak tekinthető. Sok esetben elegendő, ha csupán a késleltetés állandó értéken tartását biztosítjuk, ekkor az állandó késleltetés a számítások során korrigálható. Mindez azt jelenti, hogy elegendő $T_d = \text{állandó}$ biztosítása. A mérés során ezzel az esettel foglalkozunk.



3. ábra. A késleltetések alakulása szinkronizálatlan esetben

A 3. ábrán láthatjuk, hogy abban az esetben, amikor $T_s > T_b$, a késleltetés folyamatosan csökken, míg el nem éri a legkisebb értéket, amely T_{Send} . Az ábrán ez a T_{i-2} időpontban következik be. Ezután a késleltetés hirtelen újra megnő. Megállapíthatjuk azt is, hogy ekkor a jelfeldolgozó egység nem kap új adatot a két jelfeldolgozási időpont között, így a T_{i-1} és T_{i-2} pontban ugyanazt a T_{n-2} időpontban mintavételezett jelet használja fel. Abban az esetben, ha $T_s < T_b$, nem mintaismétlődés lép fel, hanem előfordul, hogy egy minta nem kerül feldolgozásra. Ennek oka, hogy mivel a szenzor gyorsabban szolgáltatja a mintákat, előfordul, hogy két jelfeldolgozási pont között akár két minta is érkezik, és csupán az utoljára érkezett adat kerül feldolgozásra.

Felmerülhet a kérdés, hogy a jelfeldolgozási időpontok miért nem akkor következnek be, amikor a szenzortól adatot fogad a bázisállomás, ekkor ugyanis $T_d = 0$. Az általunk vizsgált egyszerű esetben (egy szenzor és egy bázisállomás) ez valóban kivitelezhető lenne, és ekkor a késleltetés állandó maradna. Több szenzor esetén viszont már az összes szenzortól érkező adat konzisztenciáját biztosítanunk kell, és ebben az esetben a jelfeldolgozás már nem tud ilyen egyszerű módon az összes szenzorhoz egyszerre alkalmazkodni. A szenzor-bázisállomás párra a következőkben bemutatásra kerülő példák viszont akár több szenzor esetén is használható megoldást mutatnak be. Másik probléma lehet, ha a jelfeldolgozási

frekvencia adott, ekkor szintén nem alkalmazkodhat a szenzorhoz a jelfeldolgozás üteme.

Megjegyezzük, hogy T_d változási gyorsasága (az hogy milyen gyorsan nő/csökken, tehát milyen gyorsan csúsznak el egymástól a mintavételi és jelfeldolgozási időpontok) arányos a két egység órajel-generátorának hibájával. Ha tehát a T_d késleltetés T idő alatt ΔT -vel változik meg, akkor a két órajel-generátor frekvenciájának hibája, melyre a h_f jelölést alkalmazzuk, a következő módon számítható:

$$h_f = \frac{\Delta T}{T}. \quad (1)$$

A szinkronizáció tehát úgy is tekinthető, hogy $T_d = \text{állandó}$ feltétel megfelel a $\Delta T = 0$ esetnek, tehát a két egység frekvenciahibáját nullává tesszük.

A mérés során két szinkronizációs módszerrel ismerkedünk meg. Az egyik a mintavételi és jelfeldolgozási időpontok fizikai szinkronizálásán alapszik, ahol valójában is biztosítjuk $T_d = \text{állandó}$ feltételt. A másik módszer esetén mérjük a késleltetést, és jelfeldolgozási eszközökkel korrigáljuk T_d változásának hatását.

9.3.1. A részegységek fizikai szinkronizációja

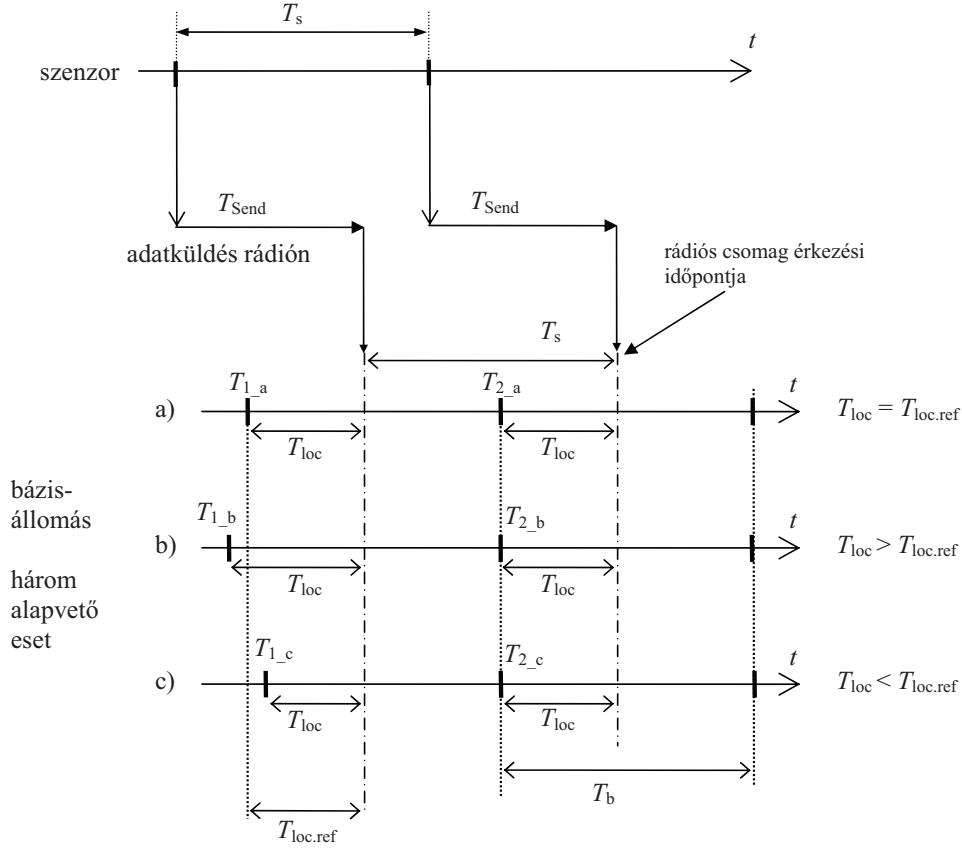
Ezen típusú szinkronizáció esetén a szinkronizálni kívánt egység óráját oly módon hangoljuk folyamatosan, hogy az átlagos időalap megegyezzen a referenciának tekintett egység időalapjával.

A konkrét alkalmazásban ez például azt jelenti, hogy úgy változtatjuk a bázisállomás jelfeldolgozási frekvenciáját (f_b), hogy az átlagos mintavételi frekvencia megegyezzen a szenzor mintavételi frekvenciájával, és ezáltal a minták keletkezése ill. feldolgozása között lévő késleltetés állandó legyen. Esetünkben ez azt jelenti, hogy a T_d értéknek, tehát a szenzor által küldött adatok érkezési idejének kell állandónak lennie, hiszen ekkor a $T_{\text{Send}} + T_d$ késleltetés (ld. 2. ábra) állandó, mivel T_{Send} állandó. A gyakorlatban – programozástechnikai okok miatt – a T_{loc} idő (ld. 2. ábra) mérése történik, de mivel $T_d = T_b - T_{\text{loc}}$, így $T_{\text{loc}} = \text{állandó}$ esetén is teljesül az állandó késleltetés feltétele. A következőkben azt az esetet tekintjük, amikor a szenzor és a bázisállomás mintavételi frekvenciái névlegesen megegyeznek.

Az $f_b = 1/T_b$ jelfeldolgozási frekvencia módosítási algoritmus a 4. ábra alapján meghatározható. Az ábrán $T_{\text{loc.ref}}$ jelöli a T_{loc} referenciaértékét, tehát azt a $T_{\text{loc}} = \text{állandó}$ értéket, amelyet tartani kell a működés során. T_{i_x} az adott esethez tartozó jelfeldolgozási időpontot jelöli, például T_{2_b} a „b)” esetben a 2. jelfeldolgozási időpont. Az ábrán látott három eset a következőknek felel meg:

- A referenciaforrás (mely esetünkben a szenzor) üzenete megfelelő időpontban érkezett, vagy legalábbis a várt időpont megfelelően kis tartományában, tehát $T_{\text{loc}} = T_{\text{loc.ref}}$ teljesül. Ebben az esetben nyilván nem kell módosítani T_b -t, hiszen ha $T_b \approx T_s$, akkor a következő rádiós üzenet is jó időpontban fog érkezni.
- A referenciaforrás üzenete később érkezett, mint az előírt érték, ezt fejezi ki $T_{\text{loc}} > T_{\text{loc.ref}}$. Ekkor a bázisállomás jelfeldolgozási időpontja a várthoz képest korábban következik be, így meg kell nyújtani a bázisállomáson a jelfeldolgozási időközt (T_b -t növelni $\rightarrow f_b$ -t csökkenteni kell), így a következő üzenet ismét a megfelelő időpontban érkezik majd.
- A referenciaforrás üzenete korábban érkezett, mint az előírt érték, ezt fejezi ki $T_{\text{loc}} < T_{\text{loc.ref}}$. Ekkor a bázisállomás jelfeldolgozási ideje le van maradva a várthoz képest, le kell csökkenteni a bázisállomáson a jelfeldolgozási időközöket (T_b -t csökkenteni $\rightarrow f_b$ -t növelni kell), így a következő üzenet ismét a megfelelő időpontban érkezik majd.

Esetünkben a fenti szinkronizációs algoritmus a bázisállomáson fut. A szinkronizálatlanság jelensége szemléltetésének érdekében ez a szinkronizáció kikapcsolható.



4. ábra. Szinkronizációs algoritmus

9.3.2. Szinkronizáció interpolációval

A 9.3.1. pontban bemutatott algoritmus abban az esetben alkalmazható, amennyiben a szinkronizálendő eszköz órája hangolható. Ez nem minden esetben lehetséges. Erre kínál egyfajta megoldást az itt bemutatásra kerülő algoritmus, melynek lényege, hogy a jelfeldolgozási időpontban becsüljük a feldolgozandó jel értékét. Ez a régebbi minták felhasználásával történhet. Ha kiszámítjuk a jel értékét a jelfeldolgozási időpontban, ez szemléletesen azt jelenti, mintha pontosan a jelfeldolgozási időpontban érkezett volna a minta, tehát $T_d = 0$ teljesül. Természetesen a becslés módszerétől függ annak pontossága, amely a késleltetés kompenzálását befolyásolja. Pontosabb becslés esetén a kompenzáció is teljesebb lesz.

A mérés során a becslésre lineáris interpolációt használunk. A módszert az 5. ábra szemlélteti. A jelfeldolgozási időpont T_i -ben következik be, d_i -vel jelöljük a szenzortól érkezett jelértékeket.

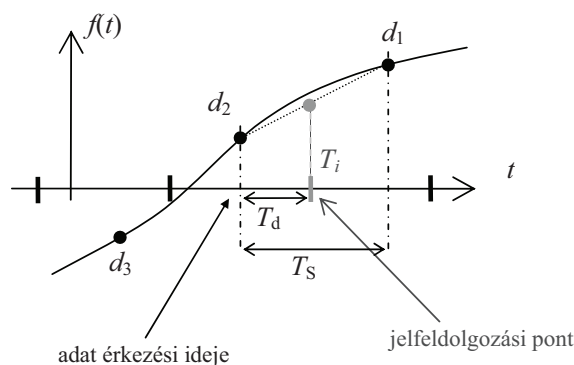
Az 5. ábra alapján egyszerű arányosságot használva kiszámíthatjuk a megfigyelt $f(t)$ jel T_i -ben felvett értékét, melyet $\hat{f}(T_i)$ -vel jelölünk (az ábrán szürke ponttal jelölt érték):

$$\hat{f}(T_i) = d_2 + \frac{d_1 - d_2}{T_s} T_d, \quad (2)$$

más formában:

$$\hat{f}(T_i) = d_1 \frac{T_d}{T_s} + d_2 \left(1 - \frac{T_d}{T_s} \right). \quad (3)$$

Megállapítható, hogy az adott időpontban felhasználjuk a megfigyelt jel előző (d_2) és következő (d_1) értékét. Mivel d_1 még nem ismert T_i -ben, így csupán a következő jelfeldolgozási időpontban számítható ki az algoritmus. Ez azt jelenti, hogy egy mintavételnyi idő késleltetést iktatunk a jelútba. Mivel ez az



5. ábra. Szinkronizáció lineáris interpolációval

érték állandó, így szinkronizációs szempontból nem okoz problémát. Láthatjuk, hogy minél régebben érkezett d_2 , annál kisebb súllyal vesszük figyelembe, ekkor ugyanis $(1 - T_d/T_s)$ közelebb kerül 0-hoz, viszont a következő d_1 mintához tartozó súly egyre nagyobb. Mivel T_d maximálisan akkora lehet, mint T_s , hiszen ennyi idő múlva már megérkezik az új minta, így $T_d/T_s \leq 1$.

Az 5. ábra alapján az is látható, hogy ugyan a becslés rendelkezik bizonyos hibával, de jobb, mintha egyszerűen a jel utolsó értékét (d_2) használnánk fel.

Figyeljük meg, hogy az interpoláció megfeleltethető olyan FIR szűrőnek, melynek együtthatóit folyamatosan változtatjuk (idővariáns). Ebből következően a becslés hibája függ T_d értékétől.

9.3.3. Gyakorlati példa a szinkronizációra

Az elméleti összefoglaló után tekintsünk egy illusztratív példát. Tegyük fel, hogy egy alkalmazott minden órában kap egy e-mailt a főnökétől, melyben megkapja a végrehajtandó feladatokat. A főnök tekinthető estünkben a szenzornak, mely adatokat küld, az e-mail kliens a bázisállomásnak, az alkalmazott pedig a feladatot végrehajtó PC-nek.

Mivel a főnök lusta és hanyag, az alkalmazott a következőhöz hasonló üzeneteket kap: „Egy óra múlva találkozunk a földszinten!”. Ugyan az alkalmazott óránként megnézi az e-mailjeit, de ha csak a legutolsó üzenetet „dolgozza fel” – és nem látja az üzenet érkezésének időpontját –, akkor az esetünkben is fennálló problémába ütközik: egy óra múlva, de mihez képest? Hasonló ez az adatok feldolgozásához: nem tudjuk mikor keletkezett az utolsó adat, így értelmezési bizonytalanságok lehetnek. Erre a problémára a bemutatott eljárások alapján két megoldás kínálkozik:

A beosztott kitapasztalja a főnök napirendjét, és alkalmazkodik (fizikailag szinkronizálódik) a főnökéhez. Tehát, ha például kitapasztalja, hogy a főnöke általában minden óra 10 perckor küld e-mailt, akkor felkészülhet arra, hogy ha minden óra 20 perckor meg tudja nézni és „fel tudja dolgozni” az e-mailt, akkor az a levél az olvasás előtt 10 perccel érkezett, és így tud mihez viszonyítani. Ehhez ki kell alakítani egy, a főnökhöz igazodó napirendet. A jelfeldolgozási példára visszatérve: tudjuk mikor érkezik az adat ($T_{loc} = \text{állandó}$ a 2. ábrán), így az időbeli értelmezésében nincs probléma.

Előfordulhat azonban, hogy a beosztott nem tud alkalmazkodni a főnökhöz, mert más jellegű kötelezettségei is vannak, vagy a főnök nem rendszeresen küldi az e-maileket. Ekkor megoldás lehet, hogy az alkalmazott beállítja e-mail kliens programját, hogy minden üzenet esetén jelezze az érkezési időt, így ha minden e-mail esetén megnézi az érkezés pontos idejét (méri T_{loc} értékét a 2. ábra szerint), akkor szintén a megfelelő döntést tudja meghozni: tudja mihez képest kell egy óra múlva találkozni.

Természetesen a jelfeldolgozásban alapvetően más problémák jelennek meg, de megállapítható, hogy ha nem tudunk a rendelkezésre álló adatokhoz időbeli információkat rendelni, az komoly problémákat okozhat.

9.4. Jelanalízis

A jelfeldolgozó rendszerek megvalósításakor a rendszer részegységeinek összehangolásán (szinkronizálásán) kívül meg kell oldanunk a megfigyelt jel feldolgozását is, így elengedhetetlen bizonyos alapvető jelfeldolgozási technikák megismerése. A villamosmérnöki gyakorlatban, főleg a jelfeldolgozásban igen nagy szerepet játszik a vizsgált jel frekvenciatartománybeli vizsgálata, spektrumának analízise. A mérés során lehetőség nyílik ezen jelanalízisben használt alapvető eszköz megismerésére és gyakorlati használatára.

A jelek spektrumának előállításához a jól ismert Fourier-transzformáció használható, amely definíció szerint a következő egyenlettel adott:

$$X(f) = \int_{-\infty}^{\infty} x(t) \cdot e^{-j2\pi ft} dt, \quad (4)$$

ahol $x(t)$ a vizsgált jel időfüggvénye, $X(f)$ a Fourier-transzformált, t az időt, f pedig a frekvenciát jelöli.

Mintavételes rendszer lévén esetünkben a fenti képlet nem használható, a mintavételezett jelek esetére definiált összefüggés a következő:

$$X(f) = T_s \sum_{n=-\infty}^{\infty} x(n) \cdot e^{-j2\pi fn}, \quad (5)$$

ahol $x(n)$ a vizsgált $x(t)$ jel mintavételezett időfüggvénye az n -edik mintavételi időpontban. Ebben az esetben f az f_s mintavételi frekvenciához képest relatív értendő frekvenciát jelöli. Mint tudjuk, a mintavételezett jelek spektruma periodikusan ismétlődik f_s periodicitással, ahol f_s a mintavételi frekvencia. Az n -edik mintavételi időpont $t = nT_s$ időpontnak felel meg, ahol $T_s = 1/f_s$ a mintavételi időköz. Természetesen (5) csupán analitikusan értékelhető ki, hiszen a $]-\infty, \infty[$ intervallumban vett szummához végtelen mennyiségű adat szükséges. Valós rendszerekben ezért a jól ismert DFT használható, mely N darab rendelkezésre álló minta felhasználásával állítja elő a spektrumot N darab pontban:

$$X(k) = \sum_{n=0}^{N-1} x(n) \cdot e^{-j\frac{2\pi}{N}kn}, \quad k = 0 \dots N-1, \quad (6)$$

ahol N a rendelkezésre álló minták számát jelöli. A mintavételi frekvencia ebben az esetben is f_s , és a DFT 0-tól f_s -ig N darab pontban egyenletesen állítja elő a spektrumot. Az $X(k)$, k -adik előállított Fourier-transzformált érték tehát $f = \frac{f_s}{N}k$ valós frekvenciának felel meg. Ez alapján számítható a DFT frekvenciafelbontása, mely Δf :

$$\Delta f = \frac{f_s}{N}. \quad (7)$$

A DFT hatékony számítására szolgál az FFT, amely bizonyos N pontszám esetén (pl. kettő valamilyen egész számú hatványának megfelelő számú minta) gyorsítja az algoritmust. MATLAB-ban ez az `fft()` függvény segítségével használható. Az `fft()` parancs nem csak ilyen speciális méretű adattömbre alkalmazható, de abban az esetben nem feltétlenül tud gyorsítási lehetőségeket biztosítani.

(6) úgy is értelmezhető, mintha (5)-ben található $x(n)$ jelet megszoroznánk egy $w(n)$ ablakkal, mely az $n = [0 \dots N-1]$ intervallumban 1, ezen kívül nulla értékű:

$$w(n) = \begin{cases} 1 & : \text{ ha } 0 \leq n < N \\ 0 & : \text{ egyébként.} \end{cases} \quad (8)$$

Ez azt jelenti, hogy nem használjuk fel a teljes jelet a spektrum kiszámításában, csupán $x(n)$ egy szeletét, amelyet az úgynevezett $w(n)$ ablakfüggvény segítségével választjuk ki. Legalapvetőbb ablakfüggvénynek az ún. *rect* ablak tekinthető, amelyet (8) írja le. Általános esetben többféle ablakfüggvényt

használhatunk fel, ekkor a DFT (6) képlete a következőképpen módosul:

$$X(k) = \sum_{n=0}^{N-1} w(n)x(n) \cdot e^{-j\frac{2\pi}{N}kn} = \sum_{n=0}^{N-1} x_w(n) \cdot e^{-j\frac{2\pi}{N}kn}, \quad k = 0 \dots N-1, \quad (9)$$

ahol $w(n)$ az ablakfüggvényt, $x_w(n)$ az ablakozott jelet jelöli. Láthattuk, hogy mivel a vizsgált jelnek csupán egy szegmensét használjuk fel, így valamilyen ablakozás mindenképpen történik, de többféle ablakfüggvény létezik. Ezen ablakfüggvények jelentőségének megismeréséhez a következőkben vizsgáljuk meg az ablakozás hatását.

Jelöljük az ablakozott $x_w(n)$ jel Fourier-transzformáltját $X_w(f)$ -fel. Mivel az ablakozott jel az eredeti jel és az ablakfüggvény szorzata: $x_w(n) = x(n) \cdot w(n)$, így az ablakozott jel spektruma az eredeti jel spektrumának és az ablakfüggvény Fourier-transzformáltjának konvolúciójával kapható meg, hiszen két jel időtartománybeli szorzata a frekvenciatartományban a két jel spektruma konvolúciójának felel meg:

$$X_w(f) = X(f) * W(f) = \int_{-\infty}^{\infty} X(f - \varphi) W(\varphi) d\varphi, \quad (10)$$

ahol a $W(f)$ a $w(n)$ ablakfüggvény spektrumát, $*$ pedig a konvolúciót jelöli.

Vizsgáljuk meg az ablakozást arra az esetre, amikor a vizsgált $x(n)$ jel egy f_0 frekvenciájú, $2A$ amplitúdójú szinuszel. Ezen jel spektruma a $\pm f_0$ frekvencián található $A\delta(f \pm f_0)$ Dirac-delta, mely természetesen mintavételezett jel esetén f_s frekvenciánként ismétlődik. Belátható, hogy $A\delta(f \pm f_0)$ és $W(f)$ konvolúciója az $A \cdot W(f \pm f_0)$, tehát az ablakfüggvény spektruma megjelenik a $\pm f_0$ frekvenciákon az amplitúdóval súlyozva (ez gyakorlatilag egy modulációnak tekinthető: az ablakfüggvényt szorozzuk egy szinuszzel, és mint tudjuk, egy f_0 frekvenciájú szinuszos jellel való szorzás az adott függvény spektrumát a $\pm f_0$ frekvenciájú pontba tolja). Ezt szemlélteti *rect* ablak esetén a 6. ábra. A szürke csúcs jelöli a jel frekvenciáján lévő Dirac-delta függvényt, tehát $X(f)$ -et, szaggatott vonal pedig az ablakfüggvény Fourier-transzformáltját, tehát $W(f)$ -et. A *rect* ablak Fourier-transzformáltja a *dsinc*($)$ (diszkrét *sinc*, hiszen a *rect* ablak is mintavételezett) függvény:

$$W_{\text{rect}}(f) = T \frac{\sin(\pi f T)}{\sin(\pi f \frac{T}{N})} = T \text{dsinc}(\pi f T), \quad (11)$$

ahol $T = NT_s$, tehát az ablakfüggvény, más szóval a regisztrátum hossza. Vegyük észre, hogy $W_{\text{rect}}(f) = 0$, ha $f = \frac{k}{T} = k \frac{1}{NT_s} = k \frac{f_s}{N} = k \Delta f$. A DFT-vel számított értékeket az ábrán körök jelölik. Ebben az esetben $\Delta f = 1$, hiszen a DFT ilyen felbontással adja meg a spektrumot, gyakorlatilag (7) szerinti gyakorissággal mintavételezzük a spektrumot.

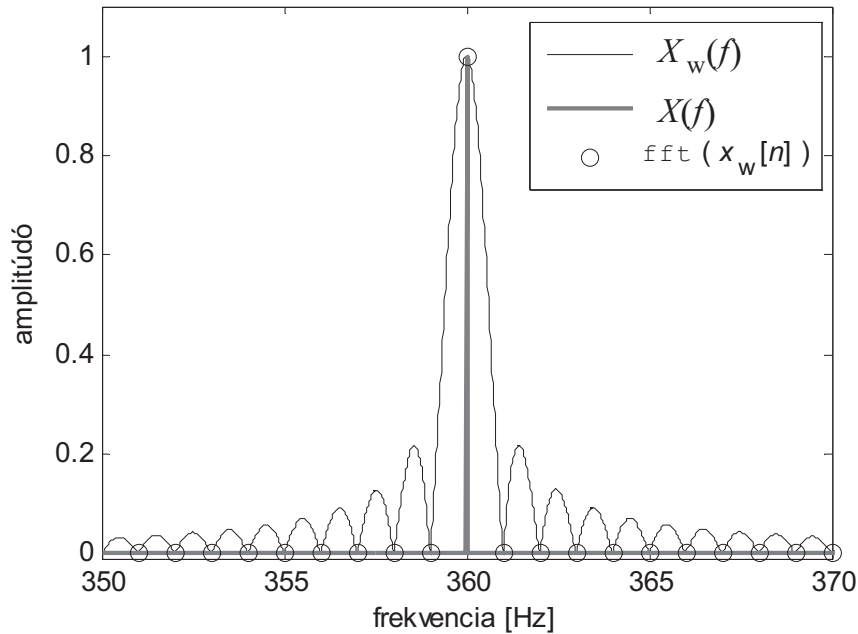
A 6. ábra azt az esetet szemlélteti, amikor a mintavételezés ún. *koherens*. Ez azt jelenti, hogy a jelből egész számú periódust dolgozunk fel, tehát a regisztrátum T hossza egész számú többszöröse (m -szerese) a jel T_j periódusidejének:

$$T = mT_j. \quad (12)$$

Mivel $T = NT_s = N/f_s$, amely (7) miatt: $T = 1/\Delta f$. A jel frekvenciája pedig: $f_j = 1/T_j$. Ezek alapján a koherens mintavételezés a következő formában is írható:

$$f_j = m\Delta f. \quad (13)$$

Tehát a DFT felbontásának a jel frekvenciája egész számú többszörösének kell lennie. Láthatjuk, hogy ebben az esetben szinuszos jelre a DFT eredménye szintén egy diszkrét Dirac-delta a vártaknak megfelelően. Ennek oka, hogy azokban a pontokban, ahol kiszámítjuk a spektrumot (körökkel jelölt pontok) az ablakfüggvény értéke nulla, kivéve a csúcspontot, ahol a jel található. Ez csak koherens mintavételezés esetén teljesül.



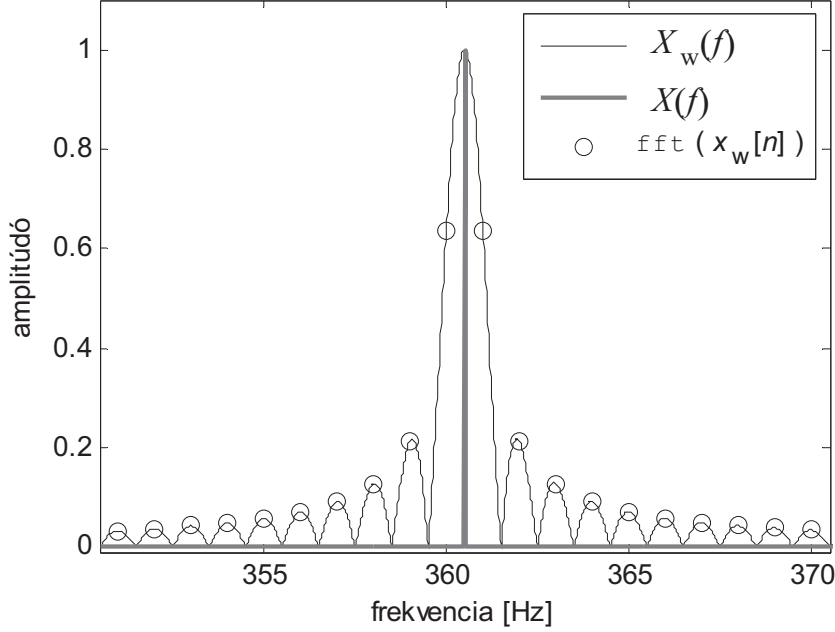
6. ábra. Ablakozott jel spektruma (*rect* ablak, koherens mintavétel)

A koherens mintavételezés a gyakorlatban nem minden esetben teljesíthető. Az, hogy a nemkoherens mintavételezés milyen problémákat okoz, a 7. ábra szemlélteti. Látható, hogy ebben az esetben a jel frekvenciája a DFT által kiszámított pontok közé esik. Ez a tény az ábrán is látható módon két problémát okoz:

- Az ablakfüggvénynek nem a zérus pontjai esnek azokra a helyekre, ahol a DFT a spektrumot számítja, így olyan frekvenciákon is megjelennek komponensek, ahol nem található jel. Ezt igen szemléletesen *spektrális szivárgásnak* (*leakage*) nevezik: az ablakfüggvény nem nulla pontjai beszivárognak a számított pontokra. Ez akkor okozhat például problémát, amikor ez a szivárgás egy kis amplitúdójú jelet elfed.
- A DFT által kiszámított spektrum csúcsának nagysága nem egyezik meg a teljes $x(n)$ jel spektrumában lévő csúcs nagyságával (szürke vonal a 7. ábrán). Ez akkor probléma, ha amplitúdót szeretnénk mérni, hiszen ekkor kisebb a mért amplitúdó, mint a jel igazi amplitúdója. Ezt a jelenséget nevezzük *tetőzésnek* (*picket fence*).

Láthattuk, hogy az ablakfüggvény miatt fellépő problémák a frekvenciatartományban szemléletesen vizsgálhatóak. Az ablakozás miatt fellépő két probléma enyhítésére az ablakfüggvénynek két feltételt kell kielégítenie:

- Minél kisebbek legyenek az ablakfüggvény Fourier-transzformáltjában az ún. oldalhullámok (szoknya), így nemkoherens mintavételezés esetén is kisebb lesz a szivárgás. Erre mutat példát a 8. ábra, melyen a DFT eredménye látható *Hanning* ablak esetén. Látható, hogy az oldalhullámok jóval kisebbek, mint az egyszerű *rect* ablak esetén.
- A tetőzés jelenségét azzal csökkenthetjük, ha az ablakfüggvény főhulláma minél kevésbé csökken. Ez azt jelenti, hogy nemkoherens mintavétel esetén a DFT által a főhullám közelében kiszámított értékek nem lesznek sokkal kisebbek, mint a jel amplitúdója. Ezt szemlélteti a 9. ábra.



7. ábra. Ablakozott jel spektruma (*rect* ablak, nem koherens mintavétel)

Ezen követelmények egymásnak némileg ellentmondóak, nehéz egyszerre mindkettőt kielégíteni. Emiatt többféle ablakozási eljárás is használatos, és a konkrét feladatnak megfelelő ablakfüggvényt kell használni. Tipikus ablakfüggvények például a hagyományos *rect* ablak, *Hanning* ablak, *flat-top* ablak. Ezen ablakfüggvények spektrumai illetve időfüggvényei a 10. ábrán láthatóak. MATLAB-ban rendre a `rectwin(N)`, `hanning(N)` és `flattopwin(N)` függvényekkel kaphatjuk meg az ablakfüggvények értékeit. Fontos dolog, hogy a torzítatlansághoz a következő feltételnek teljesülnie kell:

$$\sum_{n=0}^{N-1} w(n) = 1. \quad (14)$$

A MATLAB függvények viszont – ahogy az a 10. ábrán is látható – ezt a feltételt nem teljesítik, így a függvények által visszaadott értékeket el kell osztanunk az összegükkel, hogy (14) teljesüljön. Ne feledjük, hogy akkor is el kell végeznünk ezt a normálást, ha „nem használunk” ablakfüggvényt, hiszen ha egyszerűen vesszük egy véges hosszúságú jel DFT-jét, akkor is implicite egy *rect* ablakot használunk, tehát a DFT végeredményét le kell osztanunk N -nel, hiszen *rect* ablaknál az összes, N darab súly értéke 1, így összegük N .

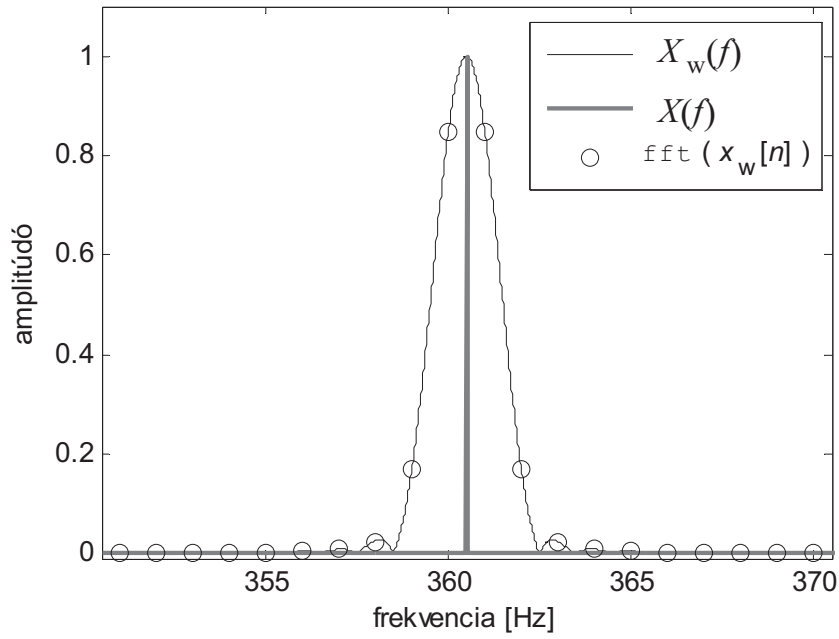
A különböző ablakok időfüggvényei a következőképpen adóttak:

Rect ablak:

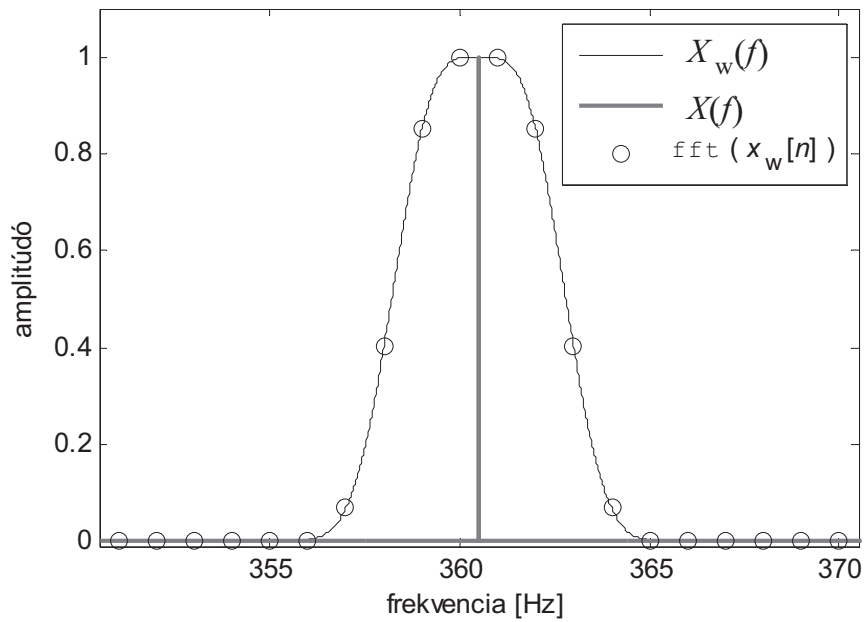
$$w(n) = \begin{cases} 1 & : \text{ha } 0 \leq n < N \\ 0 & : \text{egyébként.} \end{cases} \quad (15)$$

Hanning (más néven *von Hann*, vagy emelt koszinuszos) ablak:

$$w(n) = \begin{cases} 0,5[1 - \cos(2\pi \frac{n}{N})] & : \text{ha } 0 \leq n < N \\ 0 & : \text{egyébként.} \end{cases} \quad (16)$$



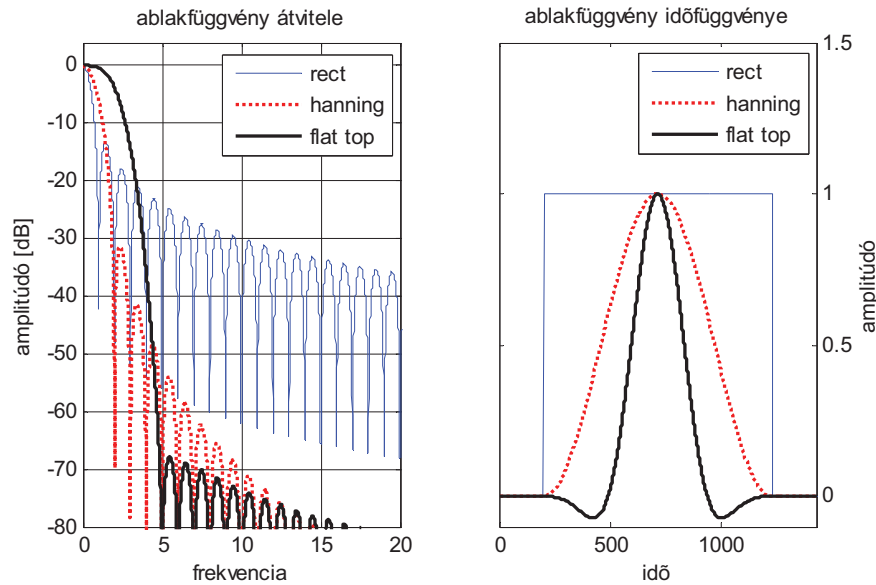
8. ábra. Ablakozott jel spektruma (*Hanning ablak*, nemkoherens mintavétel)



9. ábra. Ablakozott jel spektruma (*flat-top* ablak, nemkoherens mintavétel). Látható, hogy az `fft()`-vel számított amplitúdó alig kisebb a valódi amplitúdónál.

Flat-top ablak:

$$w(n) = \begin{cases} \sum_{i=0}^{K-1} a_i \cos(2\pi i \frac{n}{N}) & : \text{ ha } 0 \leq n < N \\ 0 & : \text{ egyébként.} \end{cases} \quad (17)$$



10. ábra. *Rect*, *Hanning* és *flat-top* ablak spektruma és időfüggvénye

9.5. A mérés során felhasznált eszközök

9.5.1. Mótok

A mérés során mitmótokat használunk. Mind a szenzor, mind a bázisállomás programja be van töltve a mótok programmemóriájába, a mérés során a mótokat nem kell átprogramozni, a különféle funkciók között a mótokon található gombok (SW) és kapcsolók (K) segítségével lehet választani. Kissé félrevezető lehet, de a mitmót I/O paneljén a gombok SW-vel vannak jelölve, így a félreértések elkerülése miatt itt is ezt a jelölést alkalmazzuk. A megvalósított funkciók a következők:

Szenzor:

K4: az AD átalakítás forrását választhatjuk ki: ON állásban a 2-es mikrofon jelét, OFF állásban a szenzorkártya Line-In bemenetét mintavételezi.

Bázisállomás:

K1: ON állásban be van kapcsolva a szinkronizáció, ekkor a 9.3.1. pontban leírt szinkronizációs algoritmus fut a bázisállomáson, és szinkronizálódik a szenzor mintavételezéséhez. OFF állásban nincs szinkronizáció.

K2: ON állásban a bázisállomás jelfeldolgozási frekvenciája névlegesen megegyezik a szenzor 1800 Hz-es mintavételi frekvenciájával. A bázisállomás ebben az esetben T_{loc} értéket továbbítja (2. és 4. ábra). OFF állásban a bázisállomás jelfeldolgozási frekvenciája 1820 Hz. Ebben az esetben a K1 állásától függetlenül nem fut szinkronizáció. A bázisállomás ebben az esetben T_d értéket továbbítja (2, 3, és 5. ábra).

SW1: A gombot megnyomva állíthatjuk a bázisállomás jelfeldolgozási frekvenciáját. K3 = ON állásban a jelfeldolgozási frekvencia 0,1 Hz-cel csökken, K3 = OFF állásban a jelfeldolgozási frekvencia 0,1 Hz-cel nő. Ezen funkció segítségével manuálisan is megvalósíthatjuk és kipróbálhatjuk a 9.3.1. pontban leírt szinkronizációs algoritmust.

SW2 / SW3: A gombokat megnyomva leállíthatjuk / elindíthatjuk a bázisállomás soros porton történő adattovábbítását.

Amennyiben rendellenes, nem várt működést tapasztalunk, a mótokon található **reset** gomb megnyomásával újraindítható az adott mót. A bázisállomás a következő formátumban továbbítja az adatokat soros porton keresztül:



A bázisállomás minden jelfeldolgozási időpontban 4 bájtos adatokat továbbít. Az első d_1 bájt a legújabb, az utolsó d_2 bájt az azt megelőző adatot jelöli (ld. 5. ábra), a középső 2 bájt pedig K_2 állásától függően T_{loc} / T_d értéket továbbítja. A d_2 adatot csupán a lineáris interpolációval történő szinkronizálás esetén használjuk fel.

Fontos megjegyezni, hogy T_{loc} / T_d értéke a mót órajelének periódusidejében értendő. Ez azt jelenti, hogy a valódi értéket úgy kapjuk, hogy a mót által küldött értéket megszorozzuk az órajel periódusidejével, mely $0,125 \mu s$, mivel a mót órajel-frekvenciája 8 MHz.



11. ábra. Adatok mentését végző program

A soros porton érkező adatok tárolásához a **SerPr** nevű program használható, ennek igen egyszerű kezelői felülete a 11. ábrán látható. Mielőtt elindítjuk a programot, kapcsoljuk be a mótot. A program elindításakor az leállítja a bázisállomáson az adatküldést, ezt az **UART START** gomb megnyomásával indíthatjuk el ismét, amint minden készen áll a mérésre. Az **UART STOP** gombbal leállítjuk a kommunikációt és kilépünk a programból. A program megjeleníti a mérés megkezdése óta eltelt időt, a fogadott adatok számát, illetve a T_{loc} / T_d értéket. Ez utóbbi a szinkronizáció vizsgálatokor fontos.

A program a beérkezett adatokat két fájlba menti. Az egyik fájl a programot tartalmazó könyvtárban **[motedata]** található **mic.dat** állomány, melyben mindig a legutóbbi adatsorozat érhető el. A **[motedata/Backups]** könyvtárban minden mérés eredményét megtalálhatjuk **mic_dátum.dat** formátumban. Ez fontos lehet, ha esetleg véletlenül újra elindítjuk a programot, mielőtt befejeztük volna a fájl feldolgozását (ekkor ugyanis felülíródik a régi állomány), illetve érdemes az adott méréshez tartozó fájlokat elmenteni, így a jegyzőkönyv íráskor ezek alapján reprodukálhatóak az eredmények. Mindkét fájlban az egymás után érkező adatok a következő formátumban találhatók meg:

d_1, T_{loc} vagy T_d, d_2

d_1, T_{loc} vagy T_d, d_2

...

9.6. Mérési feladatok

1. Jelanalízis

- 1.1. Helyezzük üzembe a mérés során használt eszközöket: kapcsoljuk be a szenzor és a bázisállomás mótót, csatlakoztassuk soros porton keresztül a bázisállomást a PC-hez. Reseteljük a bázisállomást a rajta található RESET gomb segítségével. Indítsuk el a PC-s adatgyűjtő programot, ellenőrizzük, hogy működik-e. A mérés során a szenzormót **Line-In** bemenetét használjuk, tehát kapcsoljuk K4 kapcsolót **OFF** állásba, és csatlakoztassunk egy függvénygenerátort a szenzorkártya **Line-In** bemenetére. A bázisállomáson minden kapcsoló legyen **ON** állásban. A szenzor **Line-In** bemenete AC-csatolt, így DC jel mérésére nem alkalmas, és törésponti frekvenciája kb. 30 Hz.
- 1.2. Állítsunk be a függvénygenerátoron egy tetszőleges jelet és készítsünk néhány másodperces mérést a PC-s program segítségével. A generátorral kiadott jel amplitúdója a 0 V-1,6 V tartományban legyen, ezt érdemes oszcilloszkóppal ellenőrizni. Olvassuk be a fájlokat MATLAB segítségével. (Érdemes a MATLAB-ban elvégzett műveleteket egy MATLAB script fájl írni, így egy-egy újabb adatbeolvasás és -feldolgozás egyszerűen megismételhető). Jelenítsük meg a mért jelet. Ügyeljünk mind az idő, mind az amplitúdó helyes skálázására (segítség: a szenzortól érkező adatok 8 bitesek, az AD átalakító referenciafeszültsége 3,3 V, a mintavételi frekvencia 1800 Hz).
- 1.3. Állítsunk be egy tetszőleges frekvenciájú szinuszos jelet a függvénygenerátoron. Állítsuk elő a spektrumát. Ügyeljünk mind a frekvenciatengely, mind az amplitúdó helyes skálázására. Határozzuk meg a spektrum alapján a jel frekvenciáját és amplitúdóját. A frekvenciamérés pontossága legalább 0,1 Hz legyen. Demonstráljuk az alulmintavételezés által okozott átlapolódás (*aliasing*) jelenségét, amennyiben nem tartjuk be a mintavételi tételt. Milyen frekvencián jelenik meg a mért szinuszos jel, és mekkora a valódi frekvencia? Mi a kapcsolat a kettő között?
- 1.4. Vizsgáljuk meg különböző ablakfüggvények hatását a spektrumra (*rect*, *Hanning*, *flat-top*...). Az amplitúdó tengelyen dB skálázás használata célszerű. Végezzünk méréseket a *leakage* és a *picket fence* jelenség demonstrálásához (koherens / nemkoherens mintavétel). Mekkora a spektrum kiszélesedése és a mért amplitúdó csökkenése különböző ablakok esetén? Tipp: érdemes állandó mintaszámmal dolgozni, így nem kell minden mérés során újra kiszámítani a koherens / nemkoherens mintavételezés feltételét. A pontszám változtatásával (a regiszt-rátum csonkolásával) pedig szintén beállíthatunk koherens és nemkoherens mintavételezést egy adott jelre.

2. Szinkronizációval kapcsolatos jelenségek vizsgálata

- 2.1. Indítsuk el a mérési adatokat megjelenítő PC-s programot. Vizsgáljuk meg, hogy a bázisállomás K1 kapcsolójának **ON** és **OFF** állapotában hogyan változik a T_{loc} érték (ld. 2. ábra).
- 2.2. Kapcsoljuk ki a bázisállomáson futó szinkronizációt a K1 kapcsoló **OFF** állapotba kapcsolásával. MATLAB-ban jelenítsük meg a T_{loc} érték időfüggvényét, ügyeljünk a tengelyek helyes skálázására. (Emlékeztető: T_{loc} a mót órajelének periódusidejében van megadva, az órajel frekvencia pedig 8 MHz, a mintavételi frekvencia 1800 Hz).
- 2.3. Végezzünk megfelelő hosszúságú mérést (amíg a kijelzett T_{loc} érték körülfordul, ez akár több 10 másodpercig is eltarthat). Elemezzük a mérési eredményeket! Mekkora a fűrészelés csúcsértéke? Miért? (Segítség: 3. ábra.) Melyik mót „órája” a gyorsabb? Mekkora a két mót órajel-frekvenciája közötti különbség ppm-ben?
Kiegészítő feladat: Végezzük el a mérést többféle egyenesillesztési módszerrel.
Mit jelent az a frekvenciahibára nézve, hogy lineáris az elcsúszás időfüggvénye?

- 2.4. Hagyjuk a szinkronizációt kikapcsolt állapotban: $K1 = \text{OFF}$. Végezzük el a szinkronizációt „kézzel”. Ehhez használjuk a bázisállomáson található $SW1$ nyomógombot, mellyel kis mértékben el lehet hangolni a bázisállomás jelfeldolgozási frekvenciáját a $K3$ kapcsoló által jelölt irányba. Indítsuk el a PC-s monitorozó programot, és próbáljuk meg T_{loc} értéket állandó szinten tartani. A szinkronizáció eredményét jelenítsük meg MATLAB-ban. Milyen pontos volt a szinkronizáció, azaz T_{loc} milyen intervallumban ingadozott? Kapcsoljuk be újra a szinkronizációt ($K1 = \text{ON}$, és egy új mérési regisztrátum készítésével határozzuk meg, hogy milyen pontosságú a bázisállomás automatikus szinkronizációja!

3. Szinkronizáció lineáris interpolációval

- 3.1. Változtassuk meg a bázisállomás jelfeldolgozási frekvenciáját: $K2 = \text{OFF}$ és $K1 = \text{OFF}$ állapotban legyen. Ekkor a bázisállomás jelfeldolgozási frekvenciája 1820 Hz.
- 3.2. Állítsuk be a jelgenerátor frekvenciáját tetszőleges frekvenciára, körülbelül 100-200 Hz javasolt annak érdekében, hogy mind frekvencia-, mind időtartományban megfelelően vizsgálható legyen a jelenség. Okoz-e problémát a szinkronizálatlanság alacsony frekvenciás jelek esetén, ahol lassan változik a jel?
- 3.3. Végezzük el a jel interpolációját. Ehhez az adatfájlban rendelkezésre áll a (3) képletben található összes paraméter. Jelenítsük meg közös ábrán az eredeti és az interpolált jelet. Honnan látszik a folyamatos késleltetés-változás (ld. 3. ábra)? Keressük meg a mintaismétlődés helyét. Mekkora a különbség T_s és T_b között (számítással)? Mennyi idővel változik így a késleltetés a jelfeldolgozási időpontok alkalmával? Mennyi idő alatt következik be egy teljes mintavételnyi időintervallumnyi elcsúszás? Milyen fajta modulációként jelenik meg a mért jelen?