



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

Szekeres Balázs

**ELEKTRONIKUS MÉRLEG  
SZOFTVERKOMPONENSEINEK  
TERVEZÉSE**

KONZULENS

**Orosz György**

BUDAPEST, 2013

# Tartalomjegyzék

<b>Kivonat.....</b>	<b>5</b>
<b>Abstract.....</b>	<b>6</b>
<b>1 Bevezetés .....</b>	<b>7</b>
1.1 Feladat áttekintése.....	7
1.2 Tömegmérés lehetséges módjai .....	8
<b>2 Rendszerterv.....</b>	<b>9</b>
2.1 Célkitűzések és feladatok.....	9
2.2 Hardveres felépítés .....	10
2.3 Szoftveres jelfeldolgozás és felépítés .....	13
<b>3 Felhasznált szoftverek és hardver elemek .....</b>	<b>16</b>
3.1 A mérőrendszer jelfeldolgozó egysége.....	16
3.2 Megjelenítést végző egység .....	17
3.3 VisualDSP++ .....	19
3.4 AVR Studio.....	20
3.5 Matlab .....	20
<b>4 Beágyazott eszközök szoftverkomponensei .....</b>	<b>22</b>
4.1 A jelfeldolgozó egység szoftvere.....	22
4.2 A mikrokontroller szoftvere.....	24
<b>5 Becslési módszerek.....</b>	<b>26</b>
5.1 Előzetes vizsgálatok.....	26
5.2 Rugó-erő modell .....	28
5.3 Exponenciális átlagolás.....	30
5.4 Kezdeti meredekség alapján történő súlybecslés.....	31
5.5 Amplitúdók csúcsainak átlagolása .....	31
5.6 Rendszeridentifikáció .....	36
<b>6 Becslési módszerek megvalósítása és használhatósága.....</b>	<b>40</b>
6.1 Exponenciális átlagolás.....	40
6.2 Kezdeti meredekség alapján számítás.....	41
6.3 Amplitúdók csúcsainak átlagolása .....	44
6.4 Rendszeridentifikáció .....	48
6.5 DSP-re írt szoftver online méréshez .....	50

<b>7 Összefoglalás és kitekintés.....</b>	<b>52</b>
<b>Irodalomjegyzék.....</b>	<b>53</b>

# HALLGATÓI NYILATKOZAT

Alulírott **Szekeres Balázs**, 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, 2013. 05. 26.

.....  
Szekeres Balázs

## Kivonat

A szakdolgozatom témája egy elektronikus mérleg szoftverkomponenseinek megtervezése volt. A mérlegszerkezetet készen kaptam, ezt módosítani nem volt lehetőségem. A mérőrendszer tartalmazott egy Analog Devices gyártmányú ADSP BF-537 típusú jelfeldolgozó processzort (Digital Signal Processor, röviden DSP). A feladatom ennek a rendszernek a továbbfejlesztése volt.

A dolgozat készítése során a fejlesztésnek két fő iránya volt: egy megjelenítő egység illesztése, valamint a mérendő súly becslésére egy gyors és pontos módszer kidolgozása, implementálása és tesztelése. A megjelenítőt tartalmazó eszköz rendelkezésre állt, így nekem az illesztő szoftvert kellett megvalósítani.

A becslési módszer megvalósítása során többféle módszert (egyszerű átlagolás, súlyozott átlagolás, exponenciális átlagolás, rendszeridentifikáció) is teszteltem és az általam megfelelőbbnek ítéltet a DSP-re is implementáltam. Az itt bemutatott módszerekhez kifejtem az elméleti alapokat, ezek ismertetése után pedig következnek a mérési eredmények. A szimulációk Matlab felhasználásával készültek.

## **Abstract**

The thesis presents the design of the software components of an electronic scale. The mechanical structure of the scale was complete, and I wasn't be able to modify it. The measuring system consists of an ADSP BF-537 Digital Signal Processor (DSP), manufactured by Analog Devices. My task was to improving and developing the next version of system.

The development had two main directions: the first goal was the design of software interfaces of a display device (LCD), and the second goal was the development, implementation and test of a fast and accurate method, which can predict the weight of the measured object with minimal mistake.

I have tested several predicting methods such as the simple averaging, weighted averaging, exponential averaging and system identification. I have implemented the best one on the DSP. I have presented the basic concept of every method, and after that, the results of measurements can be seen. The simulations were created by an interactive environment, the Matlab.

# 1 Bevezetés

## 1.1 Feladat áttekintése

Szakedolgozatom keretében egy mechanikus mérleg villamos jeleinek feldolgozását végző processzor szoftverét valósítottam meg. Ezen egység segítségével egyszerűen kezelhető elektronikus mérleget tudunk létrehozni, amelynek a méréstartomány 0-100 g. A rendszer központi eleme egy DSP (Digital Signal Processor), amiről a későbbiekben lesz részletesen szó. A feladat nem volt teljesen ismeretlen, mert ezzel a projekttel foglalkoztam az önálló laboratóriumi feladatom során. Az alacsony szintű jelfeldolgozás (gerjesztés, kimenet feldolgozása, nyers eredmény) már készen volt.

A szakdolgozat során ennek a rendszernek a továbbfejlesztésével foglalkoztam. A feladatom két fő részből állt. Az első rész célja egy kezelői felület megvalósítása volt, ahol meg lehet jeleníteni az eredményt és az üzemmódot. A második rész a mérési eljárás továbbfejlesztéséről szólt. A legnagyobb probléma a mérleg szerkezetének kis belső csillapítása miatti lengések jelenléte volt, amelyek miatt nagy a beállási idő. A feladatom az volt, hogy találjak egy módszert, ami a nyers mérési eredmények adataiból gyorsan és pontosan megbecsüli a lengések utáni végértéket. Az algoritmus fejlesztés offline módon történt Matlab segítségével, majd a letesztelt és működő algoritmus lett implementáltam a DSP-n.

A második fejezetben bemutatom a célkitűzéseket és a feladatokat. A harmadik témakörben a felhasznált szoftvereket és hardvereket mutatom be. Itt található részletes leírás többek között a DSP-ről. A negyedik fejezetben az általam fejlesztett szoftverek általános bemutatása olvasható. Az ötödik fejezet tartalmazza a becslési módszerek elméleti áttekintését, a hatodik pedig a gyakorlati alkalmazásukról ad információkat és következtetéseket. A hetedik és egyben utolsó fejezetben az összefoglalással és a kitekintéssel zárul a szakdolgozatom.

## 1.2 Tömegmérés lehetséges módjai

A tárgyak tömegének megmérése évezredek óta foglalkoztatja az emberiséget. Erre a célra hozták létre a mérleget<sup>[1]</sup>, majd fejlesztették tovább folyamatosan az idők folyamán.

A legrégebben használt típus a mechanikus mérlegek. Ezek két féle alapelv szerint működnek. Az első csoportba tartoznak a tömegek összehasonlításán alapulók, közülük a legismertebb a kétkarú mérleg. Kezdetben a két serpenyő tengelye felülről volt felfüggesztve, később pedig az alulról megtámasztott tengely terjedt el a nagyobb súlyok egyszerű mérése miatt. Ahogy az emberek egyre nagyobb súlyokat akartak mérni, egyre jobb mérlegeket találtak fel. Ilyen újítás során építették meg különböző karhosszúságú mérlegeket. Nagy méréstartományuk miatt rengeteg helyen tudták alkalmazni őket. Ezekre jó példa a tizedesmérleg és a hídmérleg. A második csoport alapja az erőmérés valamilyen fizikai tulajdonság változásán keresztül, például egy rugó megnyúlása. A napjainkban használt kézi mérlegek gyakran rendelkeznek egy beépített csillapítással, ahol egy majdnem zárt dugattyú légellenállása segít abban, hogy ne lengjen sokáig a mérleg a súly felhelyezése után.

Léteznek hidraulikus és pneumatikus elven működő mérlegek is. Ezen módszernél a mérendő test egy munkahengert terhel. A test súlya könnyen számítható az általa okozott nyomásnövekedés és a munkahenger aktív felületének szorzataként. Főként nagy súlyok mérésére használják.

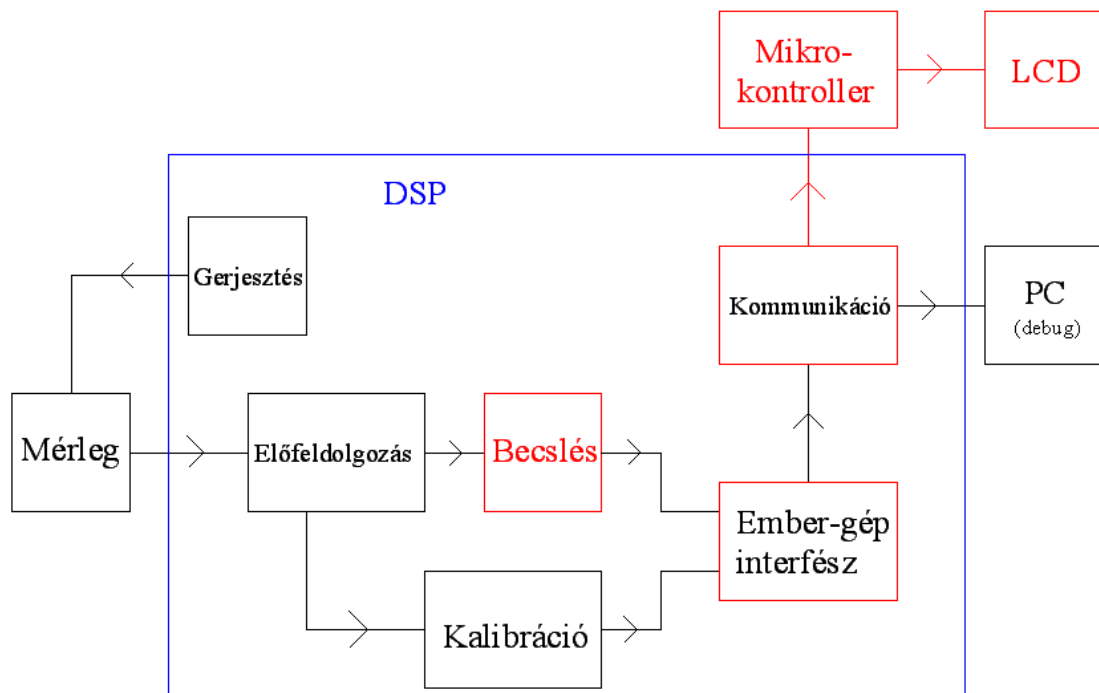
Az elektronikus mérlegek különböző érzékelők villamos jeleiből tudnak súlyt mérni. Legelterjedtebb módszer a nyúlásmérő bélyegek alkalmazása. A mérleg valamely része (általában egy rugó) a súly felhelyezésekor alakváltozást szenved, ezáltal változik a bélyeg ellenállása. A változás arányos a megnyúlással kalibráció után könnyű belőle visszszámolni a mérendő test tömegét. A megnyúlással arányos villamos jelet Wheatstone-híddal célszerű mérni. Ezt az elvet használja a rendelkezésre álló mérőrendszer is.



## 2 Rendszerterv

### 2.1 Célkitűzések és feladatok

Ebben a fejezetben ismertetem a megoldandó feladat és a megvalósítandó rendszer főbb funkcionális egységeit. A rendszer funkcionális blokkdiagramja 2.1-es ábrán tekinthető meg.



2.1 A rendszer funkciósvázlata

A 2.1-es ábrán a feketével jelölt egységek már rendelkezésre álltak a szakdolgozat készítése előtt. A piros dobozok a rendszerhez a szakdolgozat során illesztett blokkok. Azok a dobozok, amelyek pirosak, de bennük feketével írt szöveg található, azok módosítva lettek. A nagy kék dobozban található részekért a DSP felelős.

#### Megjelenítési felület

Elektronikus mérleg esetén szükség van valamilyen megjelenítő felületre, amin meg lehet tekinteni a mért értékeket. Az alapprogram soros porton küldte el ezt az értéket egy számítógépnek és ez volt a kijelző. Egy kézi mérleg esetén viszont fontos a hordozhatóság. Bár napjainkban könnyen elérhető egy laptop, ennek használata

időnként körülményes, így egy LCD (Liquid Cristal Display) kijelző tökéletes a kicsi és mobil megjelenítő szerepére. A kijelző vezérlését egy Atmel ATmega16 típusú mikrokontroller végzi, a mérést pedig a DSP. Át kell küldeni a mikrokontrollernek a mérési eredményeket vagy kalibráció esetén azt, hogy éppen hol tart a felhasználó. A kommunikáció egyirányú, a DSP csak adó a mikrokontroller pedig csak vevő. Fontos szempont hogy a kommunikáció a számértékek mellett más vezérlő karaktereket is tartalmazzon.

### **Súlybecslés gyorsítása**

A jelenleg használt mérlegszerkezet külső csillapítással nem rendelkezik, így új súly felhelyezése vagy levétele után jelentős lengések keletkeznek. A belső csillapítás miatt ezek idővel eltűnnek, de a beállási idő így 15-30 másodperc körüli érték. Bár ez nem sok, egy mérlegnél elvárás, hogy szinte egyből mutassa a felhelyezett súly nagyságát. Ebből kifolyólag érdemes beépíteni a szoftverbe valamilyen becselő algoritmust, amely ennél az időnél jóval kevesebb alatt ad nagy pontossággal egy közelítő értéket a végeredményre. A tényleges becslési eljárás mellett az algoritmusnak azt is fel kell ismernie, hogy mikor történik súlyváltozás a serpenyőben, hiszen nyugalmi helyzetben a folyamatos becslés csak felesleges erőforrás-pazarlás.

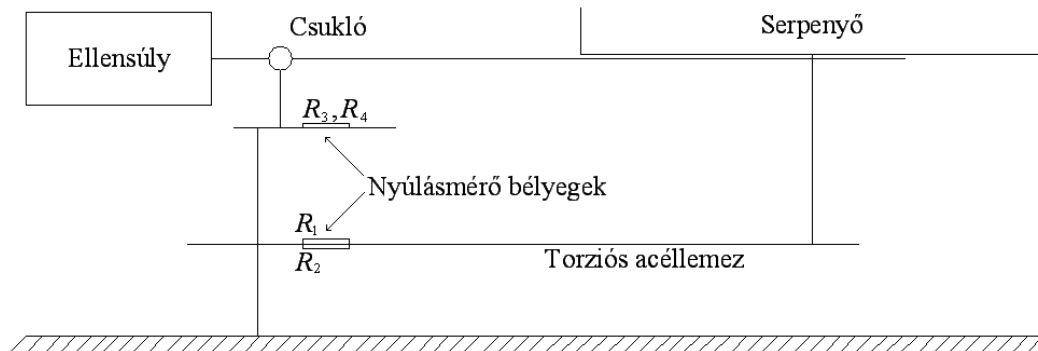
## **2.2 Hardveres felépítés**

A meglévő rendszer komplett mérést képes elvégezni, és a számított adatokat valós időben jeleníteni meg egy számítógép képernyőjén. A mérési paraméterek előre meghatározott súlyokkal kalibrálhatóak. A mérleg méréstartománya 0-100 g. Ennek a rendszernek az összeállításával az önálló laboratórium keretében foglalkoztam.

### **Mérlegszerkezet**

A mérlegszerkezet sematikus ábrája az 2.2-es ábrán látható. A mérleg mechanikai szerkezete adott, nem volt mód annak megváltoztatására. A rendszer többi komponensét illeszteni kellett a feladat támasztotta követelményekhez. Különböző magasságban helyezkednek el a nyúlásmérő bélyegek és a serpenyő az ellen súllyal.  $R_1$ ,  $R_2$ ,  $R_3$  és  $R_4$  a nyúlásmérő bélyegek. A mérés alapelve az, hogy a serpenyőre helyezett súlyok a serpenyőhöz kapcsolódó torziós acéllemez elhajlását okozzák. A lemez alsó és felső oldalán különböző nagyságú a deformáció, és ezt nyúlásmérő bélyegek segítségével mérjük. A megfelelő érzékenység eléréséhez a bélyegeket hídkapcsolásban

(Wheatstone-híd) alkalmazzuk. A serpenyő alatti függőleges rúd szerepe az, hogy akárhol helyezkedik el a mérendő súly a serpenyőn, a nyúlásmérő bélyegek lemezét csak egy pontban terheli. Ez a megoldás lehetővé teszi, hogy a súly(ok) fizikai elhelyezkedése ne befolyásolja a mérést. A terhelt lemez kis kitérés esetén nem szenved maradandó alakváltozást.



**2.2 A rendszer sematikus rajza**

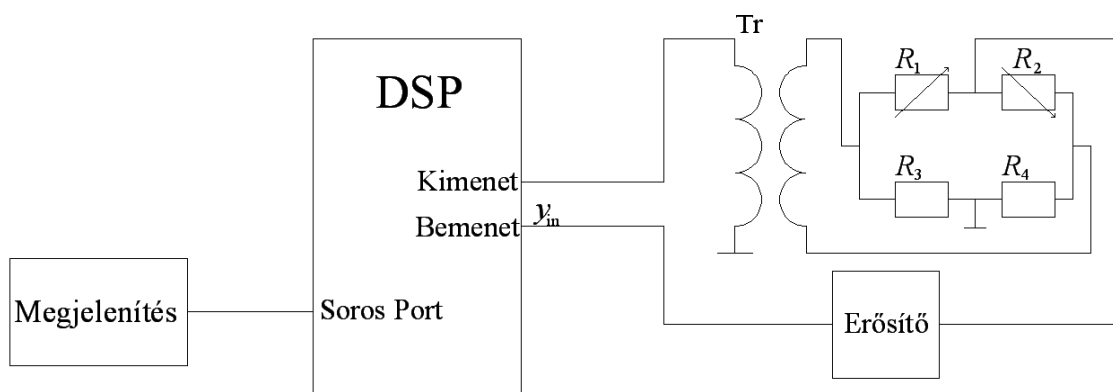


**2.3 A mérlegszerkezet fizikai képe**

### **A rendszer elrendezése**

A megvalósított rendszer blokkvázlata a 2.4-es ábrán látható. Az elrendezés központi egysége a DSP. A jelfeldolgozó egység egy Analog Devices gyártmányú ADSP-BF537-es kártya. A kártyáról bővebben a 3.1-es alfejezetben fogok írni. Ez

szolgáltatja a megfelelő hullámformájú bemeneti jelet a mérés elvégzéséhez, és ez dolgozza fel a válaszjelet is. A meghajtó jel egy leválasztó transzformátoron keresztül jut a Wheatstone-híd bemenetére. A hídkapcsolásban 4 db nyúlásmérő bélyeg van, ezekből kettő ( $R_1, R_2$ ) változtatja a hosszát a mérés során a másik kettő ( $R_3, R_4$ ) pedig a hőkompenzáció miatt van a híd másik ágában. Az érzékenység növelése érdekében a két bélyeg hosszváltozása ellentétes irányú. A jobb zajszűrés érdekében vivőfrekvenciás mérőhidat alkalmazunk. Ekkor a híd gerjesztése egy előre meghatározott frekvenciájú szinuszos jel, így előre lehet tudni a spektrum melyik része lesz hasznos jel és melyik környezeti zaj. A híd kimeneti jele egy analóg szélessávú erősítőn keresztül jut vissza a DSP-be. Erre azért van szükség mert még a fent leírt kapcsolás esetén is annyira kicsi (néhány 10 mV nagyságrendű) a jel, hogy közvetlen analóg-digitális átalakítás során nagy lenne a kvantálási zaj. A DSP kártyán történik meg a szűrés és a szűrt jel további feldolgozása, hogy a felhasználó számára is könnyen értelmezhető információt kapjunk. Az egyes elemek összekötése 50  $\Omega$ -os koaxiális kábel segítségével történik.



2.4 Rendszer blokkvázlata

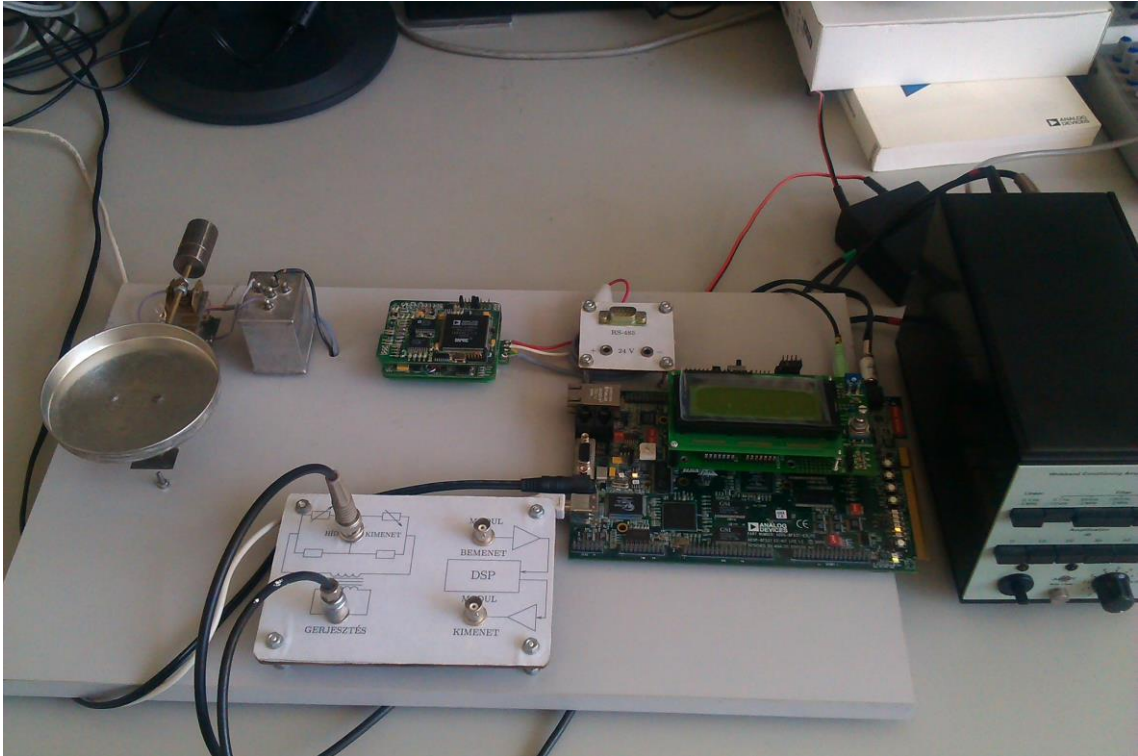
### Analóg jelkondicionálás

A rendszerhez rendelkezésre állt egy Brüel & Kjaer által gyártott 2638<sup>[2]</sup>-as típusú jelkondicionáló berendezés. Ez egy általános célú váltóáramú erősítő, a működési tartománya 0.1 Hz-től 2 MHz-ig található. Az erősítése 10 dB-essével állítható nyomógombok segítségével állítható és egy tekerő gombbal lehet még 10 dB utólagos erősítés alkalmazni 1 dB-es léptékkel. Így az eszköz maximálisan 60 dB-es erősítésre képes. Ez teljes mértékben ki van használva, mivel a hídkimenet amplitúdója nagyon kicsi.

Lehetőség van szűrés beállítására is a jelkondicionálón, 1 alul- és 4 felüláteresztő szűrő közül választhatunk. Az aluláteresztő szűrő felső határfrekvenciája

10 kHz, a felüláteresztő szűrők alsó határfrekvenciája pedig 50, 100, 200 és 400 kHz. A mérés során az aluláteresztő szűrő volt bekapcsolva, hogy minél kevesebb nagyfrekvenciás zajt kelljen kiszűrni a DSP-vel.

A készülék bemeneti impedanciája  $50\ \Omega$  vagy  $2\ M\Omega$  lehet alkalmazástól függően. Én az  $50\ \Omega$ -os variációt használtam.



2.5 A rendszer fényképe

## 2.3 Szoftveres jelfeldolgozás és felépítés

A jelfeldolgozó processzor szoftvere az egész rendszer lelke. Ez végez el minden számolást, és meghatározza a passzív elemek viselkedését is. Többféle feladatot kell ellátnia:

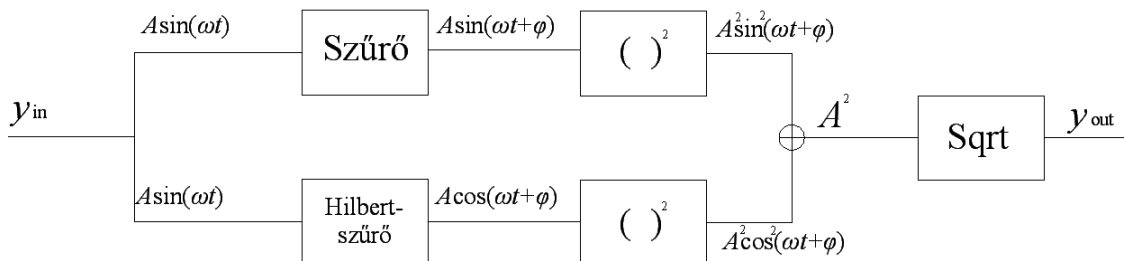
- híd gerjesztésének generálása,
- zajszűrés,
- szűrt jel további feldolgozása,
- kalibráció elvégzése,
- kommunikáció a megjelenítő eszközzel.

**Wheatstone híd gerjesztése:**

A mérés során vivőfrekvenciás mérőhidat használunk. Ekkor a híd gerjesztése valamilyen szinuszos jel. Ennek nagy előnye, hogy ismert gerjesztő frekvencia esetén tudjuk, hogy válasznak melyik része hasznos jel és melyik része zaj. DC gerjesztés esetén az erősítők offszethibája miatt nagyobb eredő offszethiba jelenne meg.

### Zajszűrés és burkoló detektálás:

Bár az erősítőben van analóg szűrés, még ezután is marad nemkívánatos zaj a bemenő jelen. Ezt digitális szűréssel igyekeztem minél jobban csökkenteni. A szűrés folyamatában blokkvázlata az 2.6-os ábrán látható:



2.6 Zajszűrés és burkolódetektálás blokkdiagramja

Az ábrán látható, hogy a szűréshez 2 db szűrőt használtam. Erre a pillanatnyi amplitúdó meghatározáshoz van szükség, mivel a mérőhid gerjesztése szinuszos, amelynek az amplitúdója (burkolója) hordozza az információt. Minél nagyobb a nyúlásmérő bélyegek deformációja, annál nagyobb a híd kimenő jelének amplitúdója. A felső egy sima sávszűrő, aminek az áteresztő tartománya a mérőhid meghajtásához használt frekvencia körül van. Az alsó egy Hilbert-szűrő, ami annyiban különbözik, hogy a szűrés során 90°-os fázistolást is végez, így a szinuszosból koszinuszos jelet készít. Szinuszos bemenő jelet feltételezve a két szűrő kimenete tehát  $A \cdot \sin(\omega t + \varphi)$  és  $A \cdot \cos(\omega t + \varphi)$ . A trigonometrikus Pitagorasz-tétel felhasználásával négyzetösszegük kiadja a jel amplitúdójának négyzetét.

$$y_{\text{out}}^2 = A^2 \sin^2(\omega t + \varphi) + A^2 \cos^2(\omega t + \varphi) = A^2 \quad (2.1)$$

Ebből az eredményből négyzetgyököt vonva megkapjuk a jel amplitúdóját.

Mivel a gerjesztőjel 6 kHz, ezért az áteresztő sáv 5850 és 6150 Hz közé került, a zárósáv pedig 5500 Hz alatt és 6500 Hz felett helyezkedik el. Ilyen paraméterek mellett egy 200-as fokszámú szűrő 42 dB-es zajelnyomást biztosít a záró tartományban.

## Skálázás és kalibráció

A híd kimenő jelének amplitúdója nyers mérési eredmény, így ezt át kell konvertálni az ember számára is könnyen felismerhető mértékegységre. A mérleg adottságait figyelembe véve automatikusan adódott, hogy a gramm a legmegfelelőbb mértékegység erre a célra.

Előzetes mérések alapján a mérleg karakterisztikája lineárisnak adódott, így a nyers eredmények konvertálása értelmezhető adatokká lineáris összefüggésként írható fel:

$$x_m = a \cdot y_{out} + b \quad (2.2)$$

$x_m$  a mért tömeg g-ban,  $y_{out}$  a nyers adat,  $a$  és  $b$  pedig konverziós paraméterek.

A mérést különböző külső hatások erősen befolyásolhatják. Ilyen befolyásoló tényező lehet az erősítő erősítése, a zaj nagysága vagy akár a híd meghajtó jelének amplitúdója. Ebből kifolyólag nem szabad az előbb leírt konverziós paramétereket konstansként beleégetni a programba. Szükséges egy kalibrációs folyamat, ahol a rendszer képes néhány előre meghatározott próbamérés alapján kiszámítani az adott feltételek esetén az illesztett görbe tengelypontját és meredekségét ( $b$  és  $a$ ). A paraméterek meghatározása a legkisebb négyzetek módszerével történik, aminek az elméleti háttere egy későbbi fejezetben lesz részletezve. A kalibrációs pontok 0 g-tól kezdve 10 g-onként helyezkednek el a 90 g-os végső értékig. Ez a felbontás lefedi a mérési tartományt és jól illeszkedő görbe számítható belőle.

Lehetett volna végpontillesztést is választani kalibrációs módszernek, ám én azért nem emellett döntöttem, mivel jóval érzékenyebb az egyszeri hibákra.

## 3 Felhasznált szoftverek és hardver elemek

### 3.1 A mérőrendszer jelfeldolgozó egysége

A jelfeldolgozáshoz az Analog Devices által gyártott ADSP-BF537 EZ-KIT<sup>[2]</sup> típusú DSP kártyát használtam. Főbb jellemzői:

- ADSP BF-537-es processzor
- 64 MB SDRAM
- 4 MB Flash memória
- Analóg audió interfész
  - 24 bites DAC és ADC
  - 48 kHz mintavételi frekvencia
  - sztereó ki- és bementi jack aljzat
  - AC csatolás
- 10/100 Mbit-es ethernet csatlakozó
- CAN csatlakozó
- 2 db UART csatlakozó
- JTAG (Joint Test Action Group) interfész
- 6 db programozható LED és 4 db programozható nyomógomb

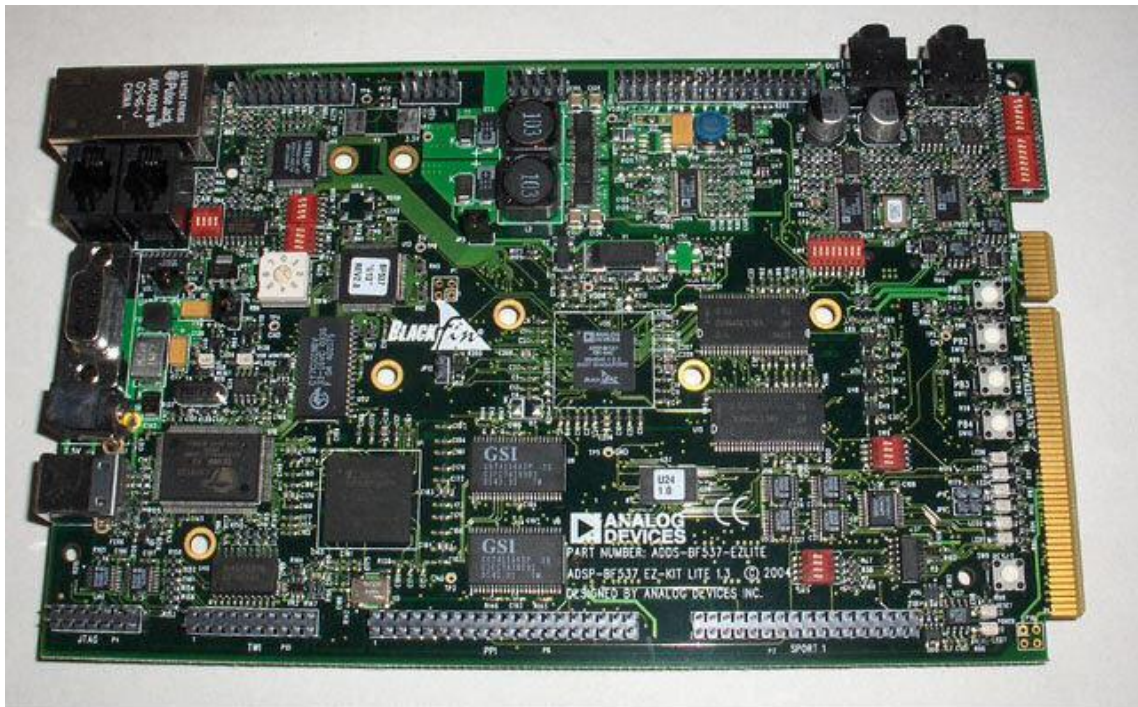
A BF-537-es jelfeldolgozó összetettsége miatt külön figyelmet érdemel. Az egység főbb jellemzői:

- Módosított Harvard architektúra
- 600 MHz belső órajel
- 133 MHz külső busz órajel
- 132 kB SRAM
- 2 db 40 bites ALU (Arithmetical and Logical Unit, aritmetikai és logikai egység)
- 2 db 16 bites MAC (Multiple and Accumulate) egység
- RISC utasításkészlet
- 182 lábas mini-BGA tokozás

Ezekkel a tulajdonságokkal ez a kártya alkalmas a feladat elvégzésére. Az audió interfész kellően nagy mintavételi frekvenciával rendelkezik a mérőhíd gerjesztéséhez és a kimenetének nyomon követéséhez. Többfajta kommunikációs lehetőség is adott a



kártya és egy számítógép vagy más beágyazott rendszer közötti adatátvitelhez. A módosított Harvard architektúrának köszönhetően a processzor egyszerre fér hozzá a program- és az adatmemóriához is. Az MAC egység segítségével egy órajel alatt el lehet végezni egy 16 bites szorzást és egy összeadást, ami a jelfeldolgozás során használt eljárásokat jelentősen felgyorsítja (pl.: digitális szűrés, Fourier-transzformáció). A nyomógombok lehetővé teszik a rendszer vezérlését a LED-eket pedig alkalmassá lehet tenni, hogy visszajelzést adjanak a program státuszáról. A 3.1-es kép egy fénykép a DSP-ről.



3.1 DSP felülnézetből

## 3.2 Megjelenítést végző egység

Ez a modul integrálva tartalmazza az LCD kijelzőt és az ezt vezérlő mikrokontrollert. A DSP-vel az UART1-es porton keresztül kommunikál, és a tápellátásáról is a jelfeldolgozó egység gondoskodik. Ezt a kártyát egy korábbi hallgató készítette a diplomamunkája során.

### LCD kijelző:

Az LCD (Liquid Cristal Display) kijelző a nevét a benne lévő folyadék kristályról<sup>[3, 971 oldal]</sup> kapta. A folyadékkristály molekulái rendezettebbek, mint a folyadékoké, de nem annyira, mint a kristályoké. Egyes folyadékkristályok a fény

polarizációs síkját elforgatják, de ezt a tulajdonságukat kis elektromos térerősség hatására elveszítik. A 90 fokos elforgatást okozó folyadékkristály lemezt két keresztezett polarizátor közé helyezik, és egy tükröt helyeznek az így kialakult rendszer mögé. Ha szemből fény éri az eszközt, akkor a 90 fokos forgatás miatt mindkét polarizátoron áthalad a fény és visszaverődve a tükrőről világosnak látszik. Ha feszültséget kapcsolunk a folyadékkristály lemezre, akkor már nem forgat 90 fokkal, így a belső polarizátor kiszűr minden fényt, sötét lesz.



**3.2 A felhasznált kijelző**

Egy korábbi hallgató projektjéből rendelkezésre állt egy EW20400LY<sup>[4]</sup> típusú kijelző. Néhány fontosabb mechanikai adat a modulról:

- Megjeleníthető karakterek: 4 db sor, soronként 20 karakter
- Modul mérete: 98\*60\*14 mm
- Kijelző területe: 76\*25,2 mm
- Karakter betűtípusa: 5\*7 pixel + kurzor

Az egység  $-20^{\circ}\text{C}$  ...  $+70^{\circ}\text{C}$  hőmérsékleti tartományban képes hosszútávon működni. 90%-os páratartalmat bír ki kondenzáció nélkül és a korrodáló anyagok ellen nem védett. A belső logika működéséhez +5 V-os tápfeszültség szükséges. Lehetőség van a kijelző kontrasztjának külső szabályzására. Választható a 4 vagy 8 adatvonalas kommunikáció, a fizikai kialakítás miatt én a 4 vonalast használtam. A kijelző háttérszíne zöld.

### **Kijelzést vezérlő mikrokontroller:**

Az LCD kijelző vezérlését egy ATmega32<sup>[5]</sup>-es mikrokontroller látja el. Ez a típus 8 bites alacsony fogyasztású mikrokontroller, amely megfelelően gyors, és elegendően nagy memóriával rendelkezik a tervezett cél megvalósításához.

A chip néhány fontosabb tulajdonsága:

- RISC architektúra
  - 131 utasítás, a legtöbb egy órajel alatt elvégezhető
  - akár 16 MIPS (millió utasítás/ másodperc) műveletvégzési képesség
  - 32\*8 db általános célú regiszter
- 16 MHz-es órajel
- 16 kB flash memória, 10000 írás/törlés ciklus élettartam
- 1 kB belső SRAM
- 512 byte EEPROM, 100000 írás/törlés ciklus élettartam
- JTAG interfész
- 2 db 8 bites időzítő/számláló, külön előosztóval és komparátor móddal
- 1 db 16 bites időzítő/számláló, külön előosztóval, komparátor és capture móddal
- 8 csatornás 10 bites ADC
- USART adó-vevő
- SPI interfész
- Programozható watchdog timer, saját oszcillátorral
- 6 féle alvó üzemmód
- Külső és belső megszakítási források
- 32 programozható I/O vonal
- Tápfeszültség: 4.5 V ... 5.5 V, L típus esetén 2.7 V ... 5.5 V

### **3.3 VisualDSP++**

A VisualDSP++ egy könnyen használható integrált szoftverfejlesztő környezet, ami az Analog Devices által gyártott Blackfin (ADSP-BFxxx), SHRAC (ADSP-21xxx) és TigerSHARC (ADSP-TSxxx) processzorcsalád tagjaihoz készült. Ezzel a fejlesztőkörnyezettel egyszerűen és hatékonyan lehet menedzselni a projekteket a kezdettől a befejezésig.

A környezet assembly, C és C++ nyelvű forráskódokat képes feldolgozni, összelinkelni, majd rátölteni a DSP-re. A futtatható programot USB-n keresztül lehet a DSP-re tölteni. A debuggoláshoz többféle lehetőség is rendelkezésre áll. Használhatunk beépített emulátort, szimulátort és JTAG-en keresztül lehet vizsgálni a processzor regisztereit, ha nem fut a program. Az emulátorral lehetőség van a processzor viselkedését tesztelni a számítógépen. A JTAG-en keresztül nemcsak megfigyelni lehet a belső regisztereket, hanem módosítani is. Lehetőség van a memória tartományokat külső fájlból feltölteni vagy értékeiket fájlba kimásolni. A kártyán található flash memóriáról is lehet bootolni ez lehetővé teszi, hogy ne kelljen mindig számítógépről felprogramozni a kártyát használat előtt.

A törtszámok egyszerű ábrázolásához rendelkezésre áll a fract16 és fract32 típus előre megírt könyvtári függvényekkel.

### **3.4 AVR Studio**

Az AVR Studio egy integrált fejlesztőrendszer, amely az Atmel AVR mikrokontrollereihez készült. A környezet támogatja az összes AVR processzorcsaládot, így egyszerűen lehet vele menedzselni mind a 8, mind pedig a 32 bites eszközhöz kapcsolódó projekteket. A rendszer rendelkezik beépített fordítóval, assemblerrel és debuggerrel, ezzel is megkönnyítve a felhasználók munkáját. A project varázslóban többféle beállítási lehetőség van a fordításhoz és a linkerhez is, majd ezek kiválasztása után el lehet kezdeni a forráskód megírását.

A fejlesztőrendszerrel JTAG-en keresztül is elérhetjük a processzort, ami jelentősen megkönnyítheti a programok debuggolását. Sajnos nekem nem álltak rendelkezésre a mikrokontroller JTAG adatvonalai a felhasznált eszközön.

### **3.5 Matlab**

A Matlab<sup>[6]</sup> egy univerzálisan használható matematikai program. Nevét a Matrix Laboratory kifejezés rövidítéséből kapta. A numerikus számításokon kívül rengeteg beépített és külső kiegészítés létezik hozzá a mérnöki és tudományos területek minden részéről. Ezen kívül rendelkezik egy magas szintű script nyelvvel algoritmus és program fejlesztéshez.

A numerikus számítási módszerek és matematikai függvények elvégzése a lehető leggyorsabb futtatáshoz lett optimalizálva. Ahogy a nevéből is látszik főleg

vektor és mátrix műveletek gyors elvégzéséhez lett a program megalkotva. Sokféle beépített funkciója van még, amik nélkülözhetetlenek a bonyolultabb problémák megoldása során. Néhány példa:

- differenciál- és integrálszámítás,
- Fourier analízis,
- lineáris egyenletrendszerek megoldása,
- ritka mátrixok feldolgozása,
- interpoláció,
- regresszió számítás.

A szoftver nagyon jó adatelemzési és megjelenítési lehetőségekkel rendelkezik. Egyszerűen be tudja olvasni az elterjedtebb szöveges vagy bináris adatfájlokat, Excel táblákat, de a különböző kép, hang vagy videó fájlokat is. A fájlokon kívül adatokat nyerhetünk ki a számítógép soros portjáról, hangkártyájáról és különböző eszközökből, mint például egy oszcilloszkóp vagy spektrumanalizátor. Az adatok ábrázolására 2-D és 3-D megjelenítés is lehetséges.

A magas szintű script nyelv sok könnyítést tartalmaz a többi programozási nyelvhez képest. Az egyik legfontosabb a mátrixokkal és vektorokkal végzett műveletek leegyszerűsítése. További hatalmas segítség, hogy nem kell a változók deklarálásával és adattípusuk kiválasztásával tölteni az időt. A memóriakezelés automatikus ezzel időt takarítunk meg és sok hibalehetőséget kerülünk el. A számos egyszerűsítés ellenére ez a program mégis tartalmazza a hagyományos programozási nyelvek legtöbb előnyös tulajdonságát mint a hibakezelés, az objektumorientált programozás vagy a saját struktúrák létrehozásának lehetősége.

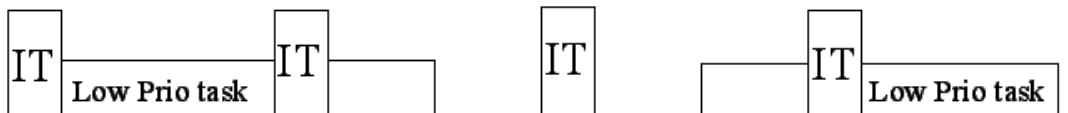
A fejlesztés során a Matlab-ot a becslési algoritmusok előzetes teszteléséhez használtam. A jelfeldolgozó egység lehetővé teszi nagy mennyiségű nyers adat gyűjtését, amelyet Matlab-ban offline módon fel lehet dolgozni. A módszer előnye, hogy a Matlab-os jelfeldolgozás egyszerűen megvalósítható és tesztelhető, így a jelfeldolgozó processzoron csak a legjobban teljesítő algoritmust kell implementálni. Mivel a beágyazott eszközökön történő szoftverfejlesztés hosszadalmas és nehézkes (PC-s fejlesztéshez képest), a fejlesztési idő ezzel a módszerrel leredukálható.

## 4 Beágyazott eszközök szoftverkomponensei

### 4.1 A jelfeldolgozó egység szoftvere

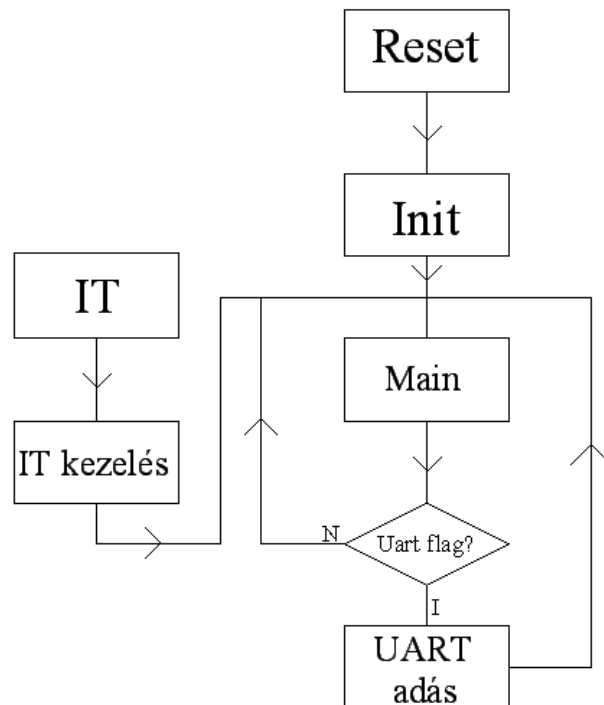
A jelfeldolgozás folyamatáról már volt szó a 2.3-as témakörben, most DSP szoftverének technikai jellegű bemutatása következik.

A DSP működése megszakítás vezérelt végrehajtású. Ez az jelenti, hogy vannak időkritikus feladatok és alacsonyabb prioritásúak. Az időkritikus feladatok (pl.: jelfeldolgozás, időzítés) egy megszakítási rutinban futnak, és az érdekében az alacsonyabb prioritású (pl.: kommunikáció, szövegformázás) feladatok ideiglenesen felfüggesztésre kerülhetnek. A folyamatok időzítési diagramja a 4.1-es ábrán látható.



4.1 DSP időzítési diagramja

### A DSP szoftverének folyamatábrája



4.2 DSP szoftverének folyamatábrája

A 4.2-es ábrán látható a DSP egyszerűsített folyamat ábrája. Az interruptkezelés (IT) doboz tartalmaz minden a jelfeldolgozáshoz kapcsolódó tevékenységet, mint a zajszűrés, burkoló detektálás, adatkonverzió, becslési algoritmus, kalibráció vagy a felvétel.

A DSP-nek 3 féle üzemmódja van közöttük a váltás nyomógombok segítségével lehetséges. Üzemmódok:

- Folyamatos mérés
- Kalibráció
- Felvétel

### **Folyamatos mérés üzemmód**

A legtöbbet használt üzemmód, ebben történik az érdemi munka, a felhasználónak szánt adat meghatározása és kijelzése.

A kimeneti jel a sztereó hangkimenet jobb csatornáján keresztül jut a mérőhídhöz. A kimeneti érték számolása során a változók fract16 típusúak. Ez egy beépített típus a VisualDSP++ fejlesztőkörnyezetben, amellyel a 16 bites tört számokat hatékonyabban lehet ábrázolni, mint a C nyelvben használatos típusokkal. A gerjesztőjelként szolgáló szinuszjel előállítás a mérési eljárás fontos része. Ez a fordító beépített könyvtári függvényeivel előállítható. A gerjesztőjel generálását a következő pszeudokód írja le, amely minden jelfeldolgozási ütemben (IT) lefut:

$$\varphi = \varphi + \Delta\varphi$$

$$\text{gerj} = \sin(\varphi)$$

$$\Delta\varphi = f/f_s, \text{ ahol } f=6000 \text{ Hz, } f_s=48000 \text{ Hz}$$

Figyelembe kell venni, hogy a DA átalakító 48000 Hz-es mintavételi frekvenciával működik, így csak 24 kHz alatti jel kiadására alkalmas.

Mivel ezek kicsi számok, a szűrés esetén a négyzetre emeléskor bizonyos részük elveszhet a számábrázolási módszer miatt. Ezen okból kifolyólag a négyzetre emelés előtt a szűrők kimenetét átkonvertáljuk float típusúvá és inentől kezdve ezzel számolunk tovább.

A szűrés során FIR-szűrőt használtam, ami bármilyen időben véges gerjesztő jelre véges választ ad. A FIR-szűrés egy diszkrét konvolúció, csak összeadásokból és szorzásokból áll, aminek számítását a DSP, az MAC utasítás gyorsasága miatt,

rendkívül hatékonyan képes elvégezni. Többfajta szűrőfokszám kipróbálása után, a hatékonyság és a processzor használat figyelembe vételével a 200 tap-es szűrő mellett döntöttem. Ilyen fokszám esetén a szűrő már jó zajelnyomással rendelkezik a záró sávban, aránylag szűk az áteresztő sávja mégis csak a processzoridő 20%-át foglalja le, így más feladatokra is jut elegendő számítási kapacitás. A sávszűrő együtthatóinak meghatározása Matlab segítségével történt.

A kommunikáció a DSP és a mikrokontroller (valamint a PC-s Debugger) között aszinkron soros porton zajlik. Az átvitel 8N1 keretű, azaz 8 adatbit után nincs paritásbit és 1 db stopbit található. Az átvitel során használt szimbólumsebesség 9600 Baud/s.

### **Kalibráció üzemmód**

Erről az üzemmódról már írtam a 2.3-as fejezetben. A kijelzőn folyamatosan követhetjük, hogy mekkora súlyt kell éppen felhelyezni. A lengés megszűnését a nyomógombokkal kell nyugtázni, majd egyből kiírásra kerül a következő mérési pont. Az utolsó kalibrációs pont után megtörténik az egyenesillesztés és automatikusan folyamatos mérés üzemmódba kerülünk, ahol már az újonnan kalibrált paraméterek felhasználásával kapott eredményeket láthatjuk.

### **Felvétel üzemmód**

Felvétel üzemmód során a külső, nagy kapacitású SD memóriában folyamatosan eltárolásra kerülnek a mért értékek. A mérés idejének hossza előre bele van égetve a programba, így megváltoztatása csak újrafordítás során történhet. Elindítása nyomógomb lenyomásával történik, leállása automatikus. Felvétel alatt is nyomon követhetőek a mért értékek a kijelzőn. Az üzemmód fő célja, hogy lehetővé tegye az adatok offline feldolgozását.

## **4.2 A mikrokontroller szoftvere**

### **Kijelző kezelése**

Az ATmega32-es mikrokontroller a DSP és az LCD kijelző közötti egység. A rendelkezésre álló eszközön úgy volt kialakítva, hogy csak ezen keresztül lehet megvalósítani a kommunikációt. Első lépésben inicializálni kell a kijelzőt. A folyamat során meg kell adni, hogy be van-e kapcsolva, 4 vagy 8 adatvonalas kommunikációt használunk, akarunk-e kurzort és ha igen villogjon-e. A megjelenítés során az alábbi függvényeket használtam:



void lcd\_init(): inicializálja a kijelzőt. 4 adatvonalas kommunikációt használtam kikapcsolt kurzorral.

void lcd\_cls (): törli a képernyő jelenlegi tartalmát minden pixelen.

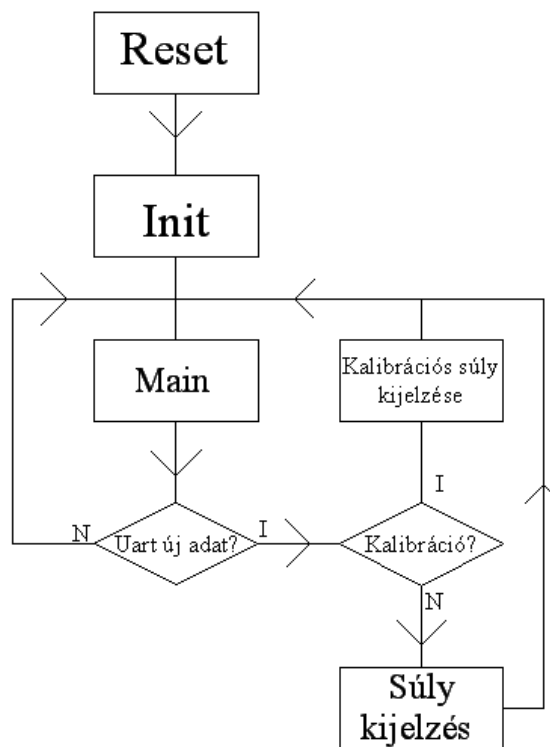
void lcd\_putchar(): kiír egy karaktert a megadott helyre.

void lcd\_putsrt(): kiír egy nullterminál stringet a megadott kezdőhelytől indulva.

### UART kezelése

A mikrokontroller a DSP-től UART-on keresztül kapta az utasításokat. Inicializálás után a szoftver folyamatosan figyeli a soros portot. Egy csomag tartalma egy vezérlő karakter, majd egy szóköz, ezt pedig egy érték követi két tizedes jegy pontossággal. A vezérlő karakter vétele után a program megjeleníti a hozzá tartozó szöveget, majd a megfelelő helyre kiírja a csomagban kapott értéket. A kiírt szöveg két féle lehet, attól függően, hogy folyamatos mérés vagy kalibrációs üzemmód van érvénybe.

### A mikrokontroller szoftverének folyamatábrája

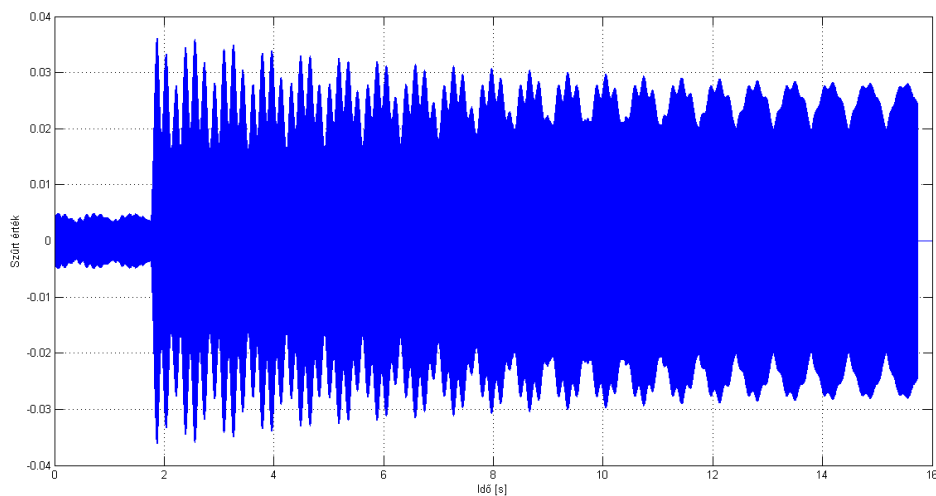


4.3 A mikrokontroller szoftverének folyamatábrája

## 5 Becslési módszerek

### 5.1 Előzetes vizsgálatok

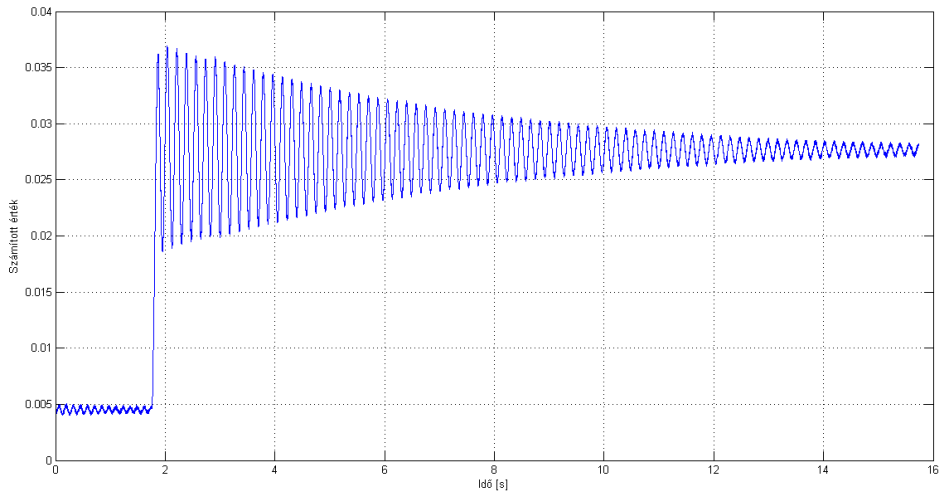
A becslési módszer megalkotása előtt meg kellett tudni, hogy mi játszódik le a mérés során, milyen jelalakokra lehet számítani. Ehhez elengedhetetlen volt néhány tesztmérés, ami alapján el lehetett kezdeni a további munkát. Először megnéztem a 2.6. ábrán látható sávszűrők kimenetét, ennek ábrázolása az 5.1-es ábrán található.



5.1 Szűrőkimenet

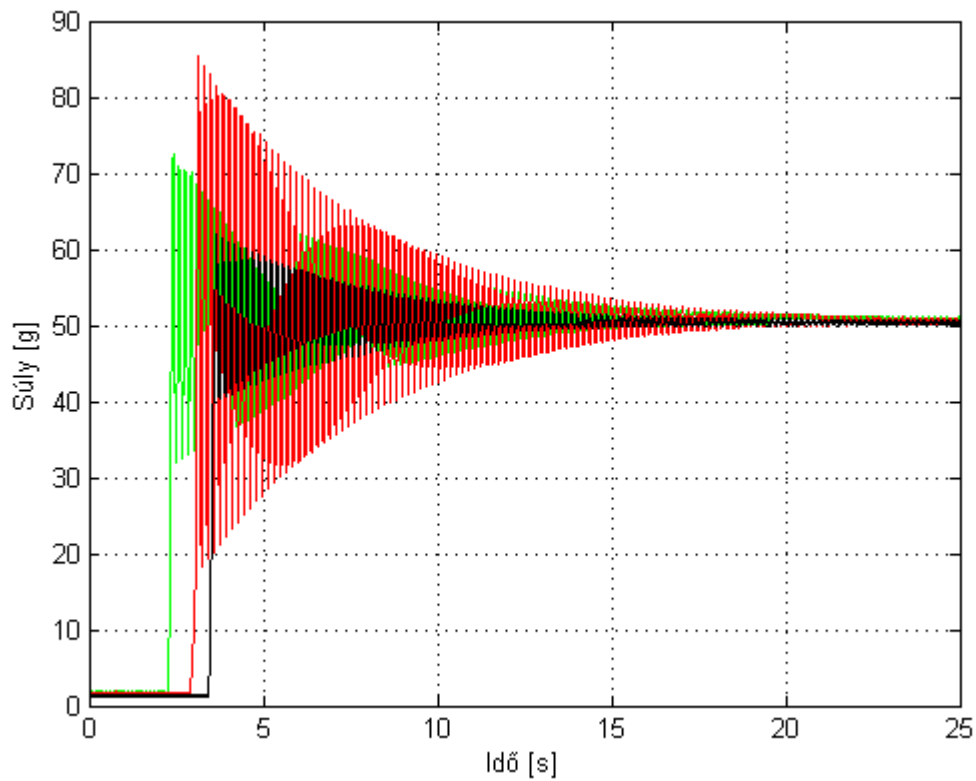
Az ábrán a szűrő kimenetének időfüggvénye látható. A mérés során egy próbasúly lett felhelyezve kb. 2 másodpercnél. Mivel csak az amplitúdót ábrázoljuk ugyanolyan a képe a sima- és a Hilbert-szűrőnek is, mivel csak a fázisuk különbözik. A fenti kép 16 másodperc felvételét mutatja, ez 768000 db float típusú szám. Ekkora adatmennyiség tárolása a DSP belső memóriájában nem lehetséges, ezért a külső SDRAM-ban lettek elhelyezve, ahonnan USB porton keresztül néhány másodperc alatt a számítógépre lehet másolni a memóriaterület tartalmát. Az y tengelyen az értékek a nyers adatok, amelyekből néhány egyszerű matematikai művelettel megkapható a grammban értelmezett tömeg a (2.2) egyenlet alapján

A két szűrő kimenetét négyzetre emelés, összeadás, majd gyökvonás után, tehát a (2.1) egyenletben leírt burkolószámítást követően, az 5.2-es ábra mutatja:



### 5.2 A burkoló időfüggvénye

A képen az  $y$  értékek még nyers mérési eredmények. Az ábrán jól látható, hogy a súly felhelyezése után egy meredek felfutás következik, majd mivel a rendszernek nincs jelentős külső csillapítása, egy szépen kirajzolódó lassú exponenciális burkolójú lecsengés a válasz.



### 5.3 Azonos súllyal végzett mérések

Az 5.3-as ábrán az  $y$  tengelyen már a grammal kiszámított súly szerepel. A képen három mérés ábrázolása található, mind a három mérés 50 g-os súly felhelyezése utáni jelalakot ábrázol, de minden alkalommal más magasságból lett felhelyezve (1-4 mm) a próbasúly a serpenyőre. A végérték minden tesztmérésnél azonos lett, de a lengések kezdeti amplitúdói nagyon széles skálán változnak.

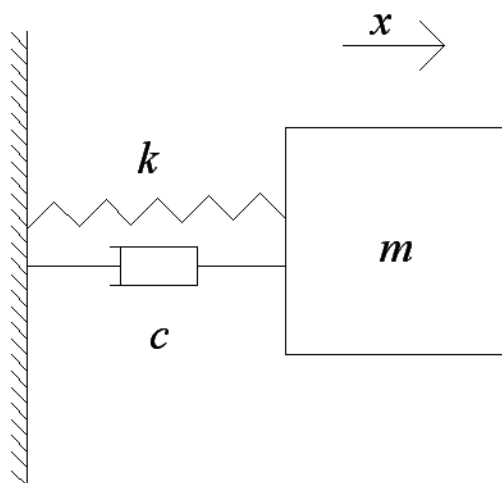
A beállítás a végértékhez 15-20 másodpercet vesz igénybe, ami meglehetősen sok. A kapott eredmények alapján szükség van valamilyen becslési eljárásra, ami a beállási időnél jóval kevesebb alatt ad egy előkalkulációt a végértékre vonatkozóan. Én az alábbi módszereket vettem számításba, amelyeket az 5.3-5.6 alfejezetekben ismertetek:

- Exponenciális átlagolás
- Kezdeti meredekség alapján számítás
- Amplitúdók átlagolása
- Rendszer identifikáció

## 5.2 Rugó-erő modell

Első megközelítésben a mérlegszerkezetét egy másodrendű lengő rendszernek tekintetem. Ebben az alfejezetben a másodrendű rendszerek legfontosabb tulajdonságait mutatom be, amelyeket a becslési módszereknél felhasználok.

Ez a modell egy olyan tömeg viselkedését írja le, ahol kimozdítás után a test nem szabadlengést végez, hanem valamilyen belső csillapítás miatt megáll egy idő után. A modell szemléltetése az 5.4-es ábrán látható:



5.4 Belső csillapítással rendelkező másodrendű lengőrendszer ábrája

A modellben szereplő  $m$  tömegű test egy  $k$  rugómerevségű rugóval és egy  $c$  belső csillapítással csatlakozik egy felülethez. A rendszert leíró differenciálegyenlet az (5.3)-as egyenlet tartalmazza<sup>[7]</sup>:

$$m \ddot{x} + c \dot{x} + kx = 0 \quad (5.3)$$

Ennek a differenciálegyenletnek a megoldását (5.4)-es képlet adja meg:

$$x(t) = A \cdot e^{-\zeta \omega t} \cdot \cos(\sqrt{1 - \zeta^2} \cdot \omega t - \phi) \quad (5.4)$$

Ahol  $A$  a kezdeti amplitúdó,  $\phi$  a kezdeti fázis,  $t$  az eltelt idő,  $\omega$  a csillapítatlan körfrekvencia, a  $\zeta$  pedig a rendszer csillapítási viszonya. A csillapítási viszony kiszámítható a differenciálegyenlet paramétereiből az alábbi képlettel:

$$\zeta = \frac{c}{2\sqrt{k \cdot m}} \quad (5.5)$$

Az egyenlet megoldásából látható, hogy az  $x(t)$  pillanatnyi kitérés értékei függenek a kezdeti amplitúdótól, a csillapítástól és az első amplitúdó óta eltelt időtől. Az időt és az amplitúdók nagyságait meg lehet határozni a mért időfüggvényből, ebből számolható lesz a  $\zeta$ . Amint megvan ez az érték  $c$  és  $k$  konstansok ismerete esetén, meg lehet határozni az  $m$  tömeget is. A  $k$  konstans meghatározható egy egyszerű méréssel. A rugómerevség mértékegysége N/m, vagyis le kell mérni, hogy ismert súly felhelyezése esetén mekkora az elmozdulása a nyúlásmérő bélyegekkkel ellátott lemez végének. A két érték hányadosa lesz a rugómerevség. A  $c$  belső csillapítás meghatározása nem egyértelmű ennyire, ugyanis nincs fizikai megfelelője a mérleghengernek, mint a  $c$ -nak, ezért visszafelé számolással egyszerűbb dolgunk van. A tesztmérések adataiból lehet számolni  $\zeta$ -t,  $k$  és  $m$  ismert, így  $c = \zeta \cdot 2\sqrt{k \cdot m}$ .

$\zeta$  számolása: az exponenciális függvénynek 3 paramétere van, de mivel én az  $e$  alapúval számolunk ez a szám 2-re redukálódik. Ezek közül az egyik a tengelypontot meghatározó szorzó ( $A$ ), a másik pedig a kitevőben lévő konstans szorzó ( $\zeta$ ), így a lecsengés függvénye, amiből  $\zeta$ -t számoljuk a  $t \rightarrow y$  leképezés esetén a következő:

$$y = A \cdot e^{-\zeta \cdot t} \quad (5.6)$$

Mivel ez egy két ismeretlenes egyenlet  $\zeta$  számításához 2 mérési pont kell. Amint ez rendelkezésre áll  $\zeta$  meghatározható az (5.7)-es egyenlet alapján:

$$\zeta = \frac{\ln\left(\frac{y_2}{y_1}\right)}{t_1 - t_2} \quad (5.7)$$

### 5.3 Exponenciális átlagolás

A lengést, mint szinuszos zavarkomponenst tekintve a becslés pontossága várhatóan javítható ezen szinuszos komponens eltávolításával, kiszűrésével. A szűrésre aluláteresztő szűrőt kell használni, mivel a hasznos jel (végérték) egy DC komponens. Első megközelítésben az aluláteresztő szűrők egyik legegyszerűbb formájával, az exponenciális átlagolóval foglalkoztam. Másik neve a „felejtő” átlagolás<sup>[8, 59. oldal.]</sup>. Ez egy súlyozott átlagolás, amely egy paraméterrel rendelkezik, az  $\alpha$ -val és számításához csak az adatsor előző adataiból kapott exponenciális átlag és az adatsor legutolsó új értéke szükséges. A számítás módja az (5.8)-as képleten látható:

$$y(n) = y(n-1) \cdot (1-\alpha) + x(n) \cdot \alpha \quad (5.8)$$

Ahol  $y(n)$  jelöli az új elem figyelembevételével kapott átlagot,  $y(n-1)$  az új elem nélküli exponenciális átlag,  $x(n)$  az új elem,  $\alpha$  pedig egy 0 és 1 közötti szám. Az egyenletbe rekurzívan visszahelyettesítve az előző elemeket a következőt kapjuk:

$$\begin{aligned} y(n) &= y(n-1) \cdot (1-\alpha) + x(n) \cdot \alpha = y(n-2) \cdot (1-\alpha) + x(n-1) \cdot \alpha^2 + x(n) \cdot \alpha = \\ &= x(0) \cdot \alpha^n + x(1) \cdot \alpha^{n-1} + \dots + x(n-1) \cdot \alpha^2 + x(n) \cdot \alpha \end{aligned} \quad (5.9)$$

A képlet kifejtéséből látható, hogy a régebben megfigyelt értékek exponenciálisan csökkenő súlyozással szerepelnek az  $n$ -dik ütemben kapott végértékben. Innen kapta a „felejtő” átlagolás nevet, hiszen ahogy haladunk előre az időben fokozatosan tűnnek el a régebbi értékek. Valójában nem felejtődnek el a régebbi megfigyelések, hiszen értékük csak a végtelenben lesz 0.

Az exponenciális átlagolás nagy előnye kicsi a memória igénye, mivel csak az előző átlagértéket és az  $\alpha$  paramétert kell tárolni. A kevés változó és paraméter felhasználásból következik, hogy a számításigény sokkal kevesebb, mint más módszerek használata esetén.

## 5.4 Kezdeti meredekség alapján történő súlybecslés

A szabadon eső tárgyak közel azonos gyorsulással indulnak el lefelé elengedés után, ezt valamennyire befolyásolja a légellenállás, de általában ettől eltekintünk. A mérlegszerkezetnek van valamekkora tehetetlensége, ez nagyban befolyásolja a kezdeti gyorsulást. Ebből kifolyólag kisebb súlyok esetén kevésbé lesz meredek a felfutási szakasz, mint a nagyobb súlyok felhelyezésekor. Ha egyszerű arányosság van a két mennyiség között egyszerűen becsülhető a súly.

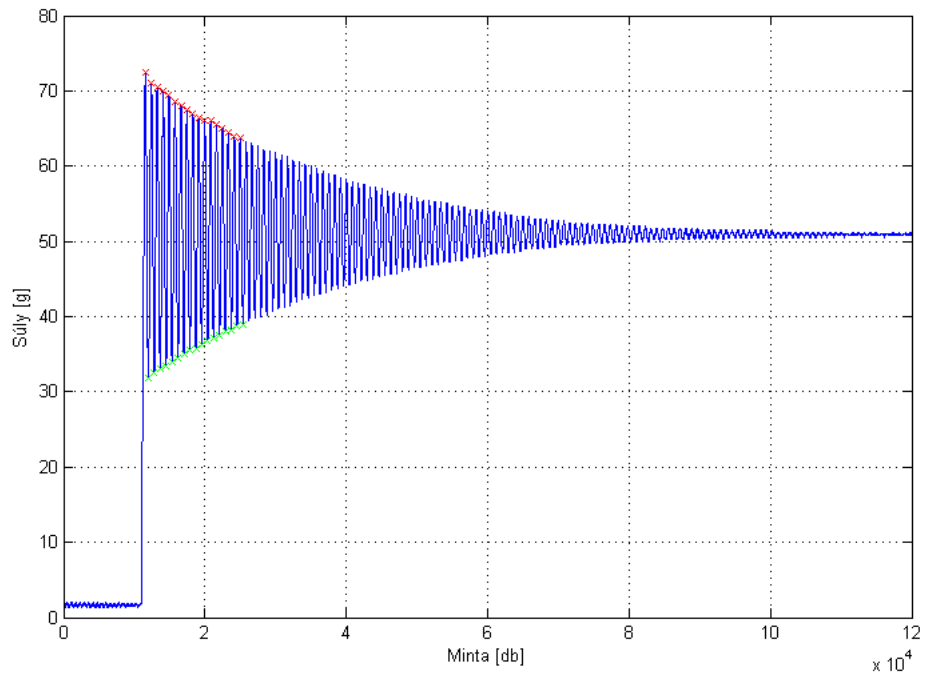
## 5.5 Amplitúdók csúcsainak átlagolása

Az 5.3 alfejezetben leírt gondolatment alapján, a mérleg kimenőjele egy végérték körül leng, és ezen végérték a lengő jel DC értéke, tehát átlaga. Ennek megfelelően az átlagolást is a végértékbecslés egy potenciális lehetőségeként vizsgáltam. Ez a módszer több lépésből épül fel. Először meg kell keresni a lokális szélső értékeket. Ezek alapján lehet számolni egyszerű átlagot, majd ennek a felhasználásával súlyozott átlagot.

### **Szélső érték keresés:**

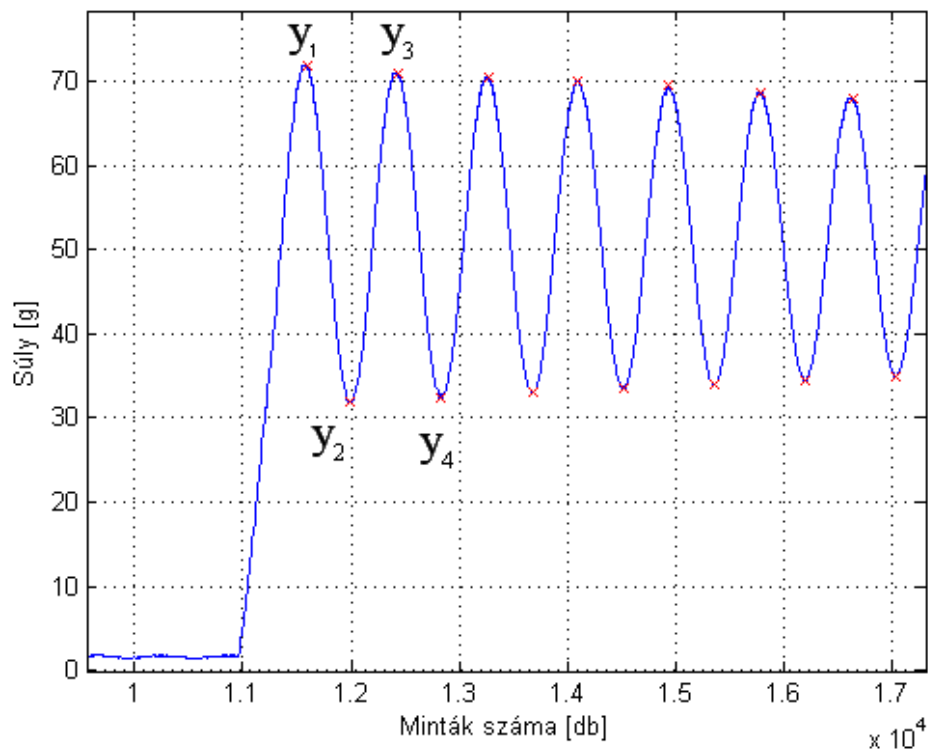
Az egyik legegyszerűbb becslési eljárás, ha egy bizonyos számú mérési eredménynek vesszük a számtani átlagát. Ehhez rengeteg adatot kellene letárolni, és a feldolgozás is hosszú időbe telne. Ebből adódik az a kézenfekvő megoldás, hogy csak olyan speciális pontokat tárolunk el, amiket könnyű megkülönböztetni a többitől, mégis jól reprezentálja az eloszlásukat. Erre a célra a legmegfelelőbbek a lokális amplitúdó maximumok és minimumok. Ezeket az akár maximum, akár minimum szélsőértékeket a továbbiakban csúcsoknak nevezem.

Az 5.5-ös ábrán láthatóak a 6.3-as alfejezetben ismertetésre kerülő algoritmus által megtalált csúcsok rárajzolva az adathalmazra, ahonnan ki lettek válogatva. A pozitívak piros színnel, a negítavak pedig zölddel vannak kiemelve.



### 5.5 Időfüggvény, pozitív és negatív amplitúdó szélsőértékekkel

Az 5.6-os ábrán látható közelről a lengés közelről. Pirossal vannak ábrázolva a csúcsok, csak ezekkel az értékekkel foglalkozok a módszer további részében.



### 5.6 Az időfüggvény eleje kinagyítva



### Egyszerű átlagolás:

Ez a módszer a megtalált pozitív és negatív csúcsok egyszerű átlagát veszi a következő módon:

$$\hat{m} = \frac{1}{N} \sum_{i=1}^N y_i \quad (5.10)$$

Az előnye, hogy rendkívül egyszerű az algoritmus és az hogy robusztus. A hátránya, hogy nem a legpontosabb az így kapott érték, mivel ez a módszer nem veszi figyelembe, hogy a pozitív és a negatív csúcsok között eltelt idő miatt a későbbi csúcs már valamivel kisebb mivel hat rá a csillapítás, tehát a pozitív és negatív csúcsok nem éjtik ki egymást tökéletesen.

Azért, hogy ezt a pontatlanságot csökkentjük súlyozott átlagolást célszerű használni.

### Súlyozott átlagolás:

Az egyszerű átlag várhatóan javítható, ha figyelembe vesszük, hogy a csúcsértékek közel exponenciálisan csökkenve követik egymást. Feltételezve, hogy az időfüggvény a végérték körül lengő exponenciálisan csillapodó szinusz, ahol a lengés amplitúdója  $A$ , a középerértéke (a becslendő végérték)  $x_0$ , az egymást követő pozitív-negatív csúcspárok értéke ( $y_i$  érték az 5.6-os ábrán jelölve) ideális esetben a következők lesznek:

$$\begin{aligned} y_1 &= x_0 + A \cdot c_1 \\ y_2 &= x_0 - A \cdot c_2 \\ y_3 &= x_0 + A \cdot c_3 \\ y_4 &= x_0 - A \cdot c_4 \end{aligned} \quad (5.11)$$

Ahol  $c_i$  a csillapítás miatti amplitúdó csökkenés, értéke:

$$c_i = e^{-\zeta(t_i - t_0)} \quad (5.12)$$

$t_0$ -at tekintjük a lengés kezdő időpontjának,  $t_i$  az  $y_i$  edik csúcshoz tartozó időpont. Algebrai átalakítások után belátható, hogy az  $x_0$  végérték a következő módon számítható:

$$x_0 = \frac{y_1 \cdot c_1^{-1} + y_2 \cdot c_2^{-1} + \dots + y_n \cdot c_n^{-1}}{c_1^{-1} + c_2^{-1} + \dots + c_n^{-1}} \quad (5.13)$$

Feltételezzük, hogy páros számú adatot dolgozunk fel, így a pozitív és a negatív tagok kiejtik egymást, mint ahogyan az (5.11) egyenletben is látható. Ez a módszer tehát a mért szélsőértékek súlyozott átlagából becsüli a végértéket. Az algoritmus nagyban épít a  $\zeta$  csillapítási tényező ismeretére, a következőkben ennek meghatározására térek ki.

A csillapítási tényező meghatározására két módszert vizsgáltam:

- $\zeta$  becslése két pontból: a 30. oldal alján és 31. oldal tetején ismertetett eljárásnak megfelelően az (5.7)-es képlet alapján. A képletben szereplő  $y_1$  és  $y_2$  értékek, valamint a  $t_1$ ,  $t_2$  időadatok a szélsőérték kereső algoritmus segítségével kaphatóak meg.
- $\zeta$  becslése több pontból: a tapasztalatok alapján két pontból történő becslés nagyon érzékeny a mérési hibákra, így teszteltem több pontra történő becslést is.

A több pont felhasználásával történő illesztés a módszerével csupán a végérték körüli lengést vizsgáljuk, ezért  $y_i$  pontokat az  $x_0$  végértékkel el kell tolni, hogy nulla körüli lengést kapjunk. Paradox módon, ehhez a mérni kívánt  $x_0$  végértékre lenne szükség, de ezen ponton annak egy kevésbé pontos módszerrel (pl. egyszerű átlag) előállított becslőjét használjuk.

Ahhoz, hogy meghatározzuk a csillapítást, az első csúcs előtti részt figyelmen kívül kell hagyni, és az egész függvényt el kell tolni úgy, hogy a középerértéke a 0-ban legyen. Ezután vesszük a függvény abszolút értékét, ezzel művelettel a minimum csúcsok átkerülnek a pozitív félsíkra. Ekkor a kapott értékek a csúcsok távolságát adják meg a középerértéktől számítva. Ezek az értékek időben előrehaladva exponenciálisan csökkennek, és a csökkenés időállandója az ismeretlen  $\zeta$  paraméter. A feladat tehát egy paraméterbecslés, ami a mérési pontokra történő exponenciális függvény illesztésével oldható meg. Exponenciális görbeillesztést érdemes úgy megvalósítani, hogy vesszük az adatok logaritmusát:

$$Ae^{-\zeta t} \rightarrow \ln(Ae^{\zeta t}) = -\zeta \cdot t \cdot \ln(e) + \ln(A) = -\zeta \cdot t + \ln(A) \quad (5.14)$$

ahol  $\zeta$  a csillapítási tényező,  $t$  az idő vagy a mintaszám (a kettő között csak egy konstans szorzó van),  $A$  a kezdeti csúcserték. Látható, hogy egy lineáris függés van a logaritmus

elvégezése után, mégpedig egy konstans értékről csökken az időfüggvény, aminek a meredeksége éppen az exponenciális csillapítási tényező. Az ekképpen transzformált adatokból  $\zeta$  lineáris regresszióval meghatározható.

A regressziószámításhoz a legkisebb négyzetek módszerét használtam. Az eljárás onnan kapta a nevét, hogy a használat során az eltérések négyzetösszegét igyekszünk minimalizálni. Tetszőleges fokszámú polinom illeszthető a mérési adatokra, a jelenlegi esetben az első fokú (egyenes) elég pontos.  $x \rightarrow y$  leképezés esetén az  $y=f(x)$  képletben az  $x_i$  értékek a mérési pontok az  $y_i$ -k pedig a mért értékek. Erre az adathalmazra akarunk olyan  $\hat{y}=f(x)$  függvényt illeszteni, ahol a kumulált kvadrátikus hiba ( $Q$ ) minimális. A hiba számítása:

$$Q = \sum_{i=1}^n \left( \hat{y}_i - y_i \right)^2 \quad (5.15)$$

Első fokú közelítés esetén  $\hat{y}=ax+b$ , ahol  $a$  az egyenes meredeksége és  $b$  a tengelypontja. A hiba csak akkor lehet minimális, ha minden paraméter ( $a, b$ ) szerinti parciális derivált nulla.

$$\frac{\partial Q}{\partial b} = 0 \quad \text{és} \quad \frac{\partial Q}{\partial a} = 0 \quad (5.16)$$

Behelyettesítve az  $\hat{y}$ -ot  $Q$ -ba és elvégezve a deriválást az alábbi eredményeket kapjuk:

$$\frac{1}{2} \frac{\partial Q}{\partial b} = \sum_i^n (ax_i + b - y_i) = a \sum_i^n x_i + nb - \sum_i^n y_i = 0 \quad (5.17)$$

$$\frac{1}{2} \frac{\partial Q}{\partial a} = \sum_i^n (ax_i + b - y_i)x_i = a \sum_i^n x_i^2 + b \sum_i^n x_i - \sum_i^n x_i y_i = 0 \quad (5.18)$$

Ezekből az egyenletekből  $a$ -t és  $b$ -t kifejezve:

$$a = \frac{\sum_{i=1}^n (x_i y_i) - \frac{\sum_{i=1}^n y_i}{n} \sum_{i=1}^n x_i}{\sum_{i=1}^n x_i^2 - \frac{\left( \sum_{i=1}^n x_i \right)^2}{n}} \quad \text{és} \quad b = \frac{\sum_{i=1}^n y_i - m \sum_{i=1}^n x_i}{n} \quad (5.19)$$

Amint kiszámítottuk a meredekséget, tudjuk számolni a csillapítást is az (5.20)-as egyenlet alapján:

$$\zeta = e^a \quad (5.20)$$

A részletszámítások közlését követően, a súlyozott átlagolásra kidolgozott módszer a következő módon foglalható össze:

- 1.)  $y_i$  szélsőértékek mérése, hozzájuk tartozó  $x_i$  időadatok mentése
- 2.)  $x_0$  első becslése (5.10) alapján
- 3.)  $y_i = y_i - x_0$  eltolás
- 4.)  $y_i = \ln(y_i)$  logaritmus transzformáció végrehajtása az eltoló pontokon
- 5.) lineáris regresszióval  $x_i$  és  $y_i$  párosokból (5.19) és 5.(20) alapján  $\zeta$  becslése
- 6.) (5.12) alapján  $c_i$  számítása
- 7.) (5.13) alapján  $x_0$  számítása az eredeti  $y_i$  értékek alapján.  $x_0$  tekinthető a becsült tömegnek.

## 5.6 Rendszeridentifikáció

Az identifikáció során meghatározzuk a rendszer empirikus modelljét. Azt feltételeztem, hogy a rendszer egy másodrendű lengőtag jól közelíti. Ennek az átviteli függvénye:

$$H(s) = \frac{A_0}{1 + 2\zeta \frac{s}{\omega_0} + \frac{s^2}{\omega_0^2}} \quad (5.21)$$

A képletben  $A_0$  a körerősítés,  $\zeta$  a csillapítás és  $\omega_0$  pedig a csillapítatlan körfrekvencia. Vegyük úgy, hogy a rendszer gerjesztése az egységugrás és a körerősítés pedig a felhelyezett súly nagyságával arányos, így  $A_0$ -at kell meghatározni.

Ez az átviteli függvény folytonos időben írja le a rendszert, de az AD átalakító 48 kHz-es mintavételi frekvenciával digitalizálja be a jelet, így diszkrét idejű lesz a feldolgozás. Folytonos időből a bilineáris transzformáció felhasználásával alakíthatjuk a rendszerünket diszkrét idejűvé <sup>[9, 441 oldal]</sup>. Ezt a következő helyettesítéssel tudjuk elvégezni:

$$s = \frac{2}{T_s} \cdot \frac{z-1}{z+1} \quad (5.22)$$

A (5.22)-es egyenletet behelyettesítve az (5.23)-esbe, a következőt kapjuk:

$$H(s) \rightarrow H(z) = \frac{A_0}{1 + \frac{2\zeta}{\omega_0} \cdot \frac{2}{T_s} \cdot \frac{1-z^{-1}}{1+z^{-1}} + \frac{1}{\omega_0^2} \cdot \frac{4}{T_s^2} \cdot \frac{(1-z^{-1})^2}{(1+z^{-1})^2}} \quad (5.23)$$

Használjuk fel a következő konstans jelöléseket:

$$k_1 = \frac{2\zeta}{\omega_0} \cdot \frac{2}{T_s} \quad \text{és} \quad k_2 = \frac{1}{\omega_0^2} \cdot \frac{4}{T_s^2} \quad (5.24)$$

Helyettesítsük be ezeket, majd szorozzuk be a tört számlálóját és nevezőjét is  $(1+z^{-1})^2$ -el. Ezzel nem változtatjuk meg az értékét, mivel összességében 1-el szorozzuk meg.

$$H(z) = \frac{A_0 \cdot (1+z^{-1})^2}{(1+z^{-1})^2 + k_1 \cdot (1+z^{-1})(1-z^{-1}) + k_2 \cdot (1-z^{-1})^2} \quad (5.25)$$

A zárójeleket felbontva, majd a tagokat fokszám szerint rendezve a következőhöz jutunk:

$$H(z) = A_0 \frac{1+2z^{-1}+z^{-2}}{1+k_1+k_2 + (2-2k_2) \cdot z^{-1} + (1-k_1+k_2) \cdot z^{-2}} \quad (5.26)$$

Ahhoz, hogy ezt az átviteli függvényt a megszokott formára hozzuk a nevezőből ki kell emelni  $(1+k_1+k_2)$ -t.

$$H(z) = \frac{A_0}{1+k_1+k_2} \frac{1+2z^{-1}+z^{-2}}{1 + \frac{(2-2k_2)}{1+k_1+k_2} \cdot z^{-1} + \frac{(1-k_1+k_2)}{1+k_1+k_2} \cdot z^{-2}} \quad (5.27)$$

Vezessük be a következő paramétereket:

$$l_1 = \frac{(2-2k_2)}{1+k_1+k_2}, \quad l_2 = \frac{(1-k_1+k_2)}{1+k_1+k_2} \quad \text{és} \quad A_1 = \frac{A_0}{1+k_1+k_2} \quad (5.28)$$

Behelyettesítve ezeket a paramétereket sokkal kezelhetőbb alakú lesz az átviteli függvény. Az átviteli függvény a válaszjel és a gerjesztés hányadosa.

$$\frac{Y}{U} = H(z) = A_1 \frac{1+2z^{-1}+z^{-2}}{1+l_1 \cdot z^{-1} + l_2 \cdot z^{-2}} \quad (5.29)$$

A keresztszorzás elvégezve a következő rekurzív összefüggést kapjuk  $y$   $n$ -edik ütemére:

$$y(n) = A_1 \cdot u(n) + 2A_1 \cdot u(n-1) + A_1 \cdot u(n-2) - l_1 \cdot y(n-1) - l_2 \cdot y(n-2) \quad (5.30)$$

A célunk  $A_1$ ,  $l_1$  és  $l_2$  paraméter meghatározása. Mátrixos formában felírva ezt az egyenletet, az összes időpontra:

$$\mathbf{y} = \mathbf{U} \cdot \mathbf{p} \quad (5.31)$$

Az (5.31) egyenletben  $\mathbf{y}=[y(0) \ y(1) \ \dots \ y(n)]^T$  tartalmazza a rendszer mért kimenetét,  $\mathbf{p}=[A_1 \ l_1 \ l_2]^T$  tartalmazza a becsülni kívánt paramétereket,  $\mathbf{U}$ -t pedig az időben eltoltt  $u(n)$  és  $y(n)$  értékek alkotják. A mérési zajok és modellezési hibák miatt egzakt egyenlőség nem írható fel, de az eltérés négyzetes értelemben minimalizálható, ha  $\mathbf{p}$ -t a következő módon választjuk:<sup>[9, 384]</sup>:

$$\mathbf{p} = (\mathbf{U}^T \cdot \mathbf{U})^{-1} \cdot \mathbf{U}^T \cdot \mathbf{y} \quad (5.32)$$

Ez a módszer  $\mathbf{p}$  számolására lényegében egy többváltozós regresszió számolás, ahol  $\mathbf{U}$  tartalmazza a magyarázó változók vektorait és  $\mathbf{y}$  az eredmény vektor. A Matlab ezen modellparaméterek meghatározására tartalmaz beépített függvényt, pl. a System Identification toolbox *arx()* parancsát. Azért ismertetem részletesen a paraméterbecslést, mert szem előtt kell tartanom, hogy amennyiben a módszer ígéretes, implementálni kell a jelfeldolgozó processzoron, ez pedig az eljárás alaplépéseinek ismeretét.

Az identifikációhoz szükséges  $u(n)$  gerjesztés ismerete. Itt feltételeztem, hogy a gerjesztés egységugrás:  $u(n) = 1$ , ha  $n \geq 0$  és  $u(n) = 0$ ,  $n < 0$ , tehát a súly felhelyezését követően egyenletes a terhelés. A ráhelyezett súly tömegét nem a gerjesztésbe, hanem a rendszer körerősítésébe foglalom bele, így matematikailag egyszerűen kezelhető a probléma, hiszen a tömeg ismeretlen, így célszerű azt az egyébként is meghatározandó rendszerparaméterekbe belefoglalni, nem pedig a gerjesztésbe.

Amint az  $A_0$ ,  $l_1$  és  $l_2$  paraméterek ismertek, a végérték tétel<sup>[9, 440 oldal]</sup> segítségével meghatározható, hogy a rendszer kimenete milyen értéket vesz fel állandósult állapotban, ha a gerjesztés egység ugrás:

$$\hat{m}_{ki} = \lim_{z \rightarrow 1} \left( (1 - z^{-1}) \cdot F(z) \right) = \lim_{z \rightarrow 1} \left( (1 - z^{-1}) \cdot \frac{1}{(1 - z^{-1})} \cdot H(z) \right) =$$

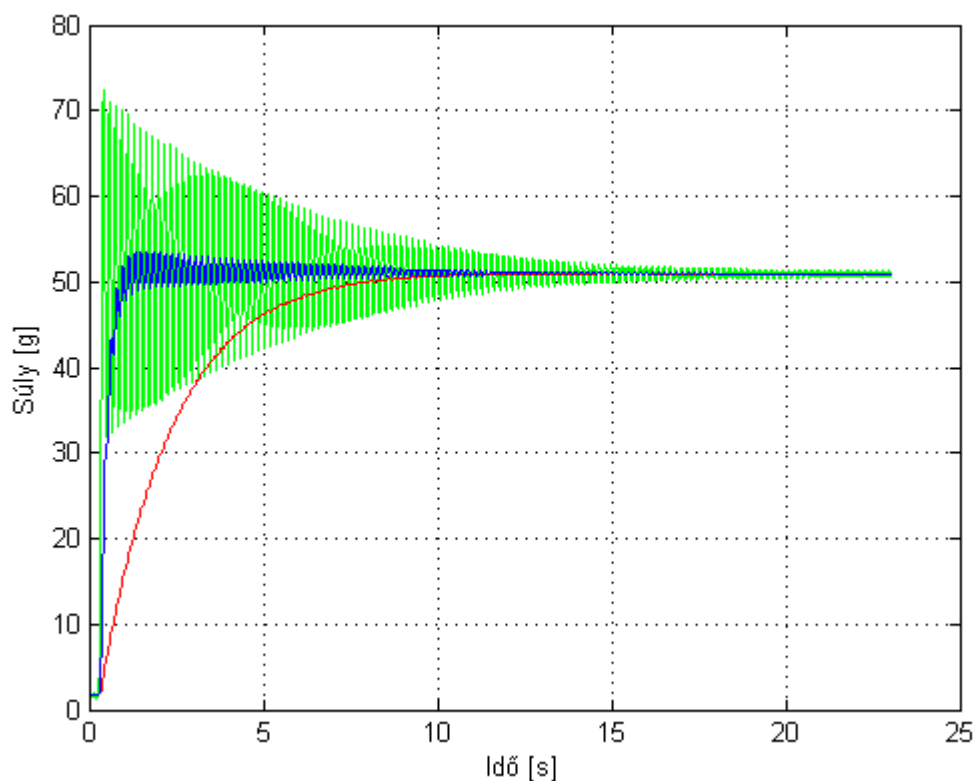
$$= \lim_{z \rightarrow 1} (H(z)) = A_1 \cdot \frac{4}{1 + l_1 + l_2} \quad (5.33)$$

## 6 Becslési módszerek megvalósítása és használhatósága

Ebben témában az 5. fejezetben ismertetett becslési módszerek implementálása és tesztelése olvasható.

### 6.1 Exponenciális átlagolás

Az előző fejezetben említett  $y(n) = y(n-1) \cdot (1-\alpha) + x(n) \cdot \alpha$  képlet használata esetén kettő változót kell tárolni, az  $y(n-1)$ -et és az  $\alpha$  paramétert. Az exponenciális átlagolás alkalmazását az 6.1-as ábra mutatja:



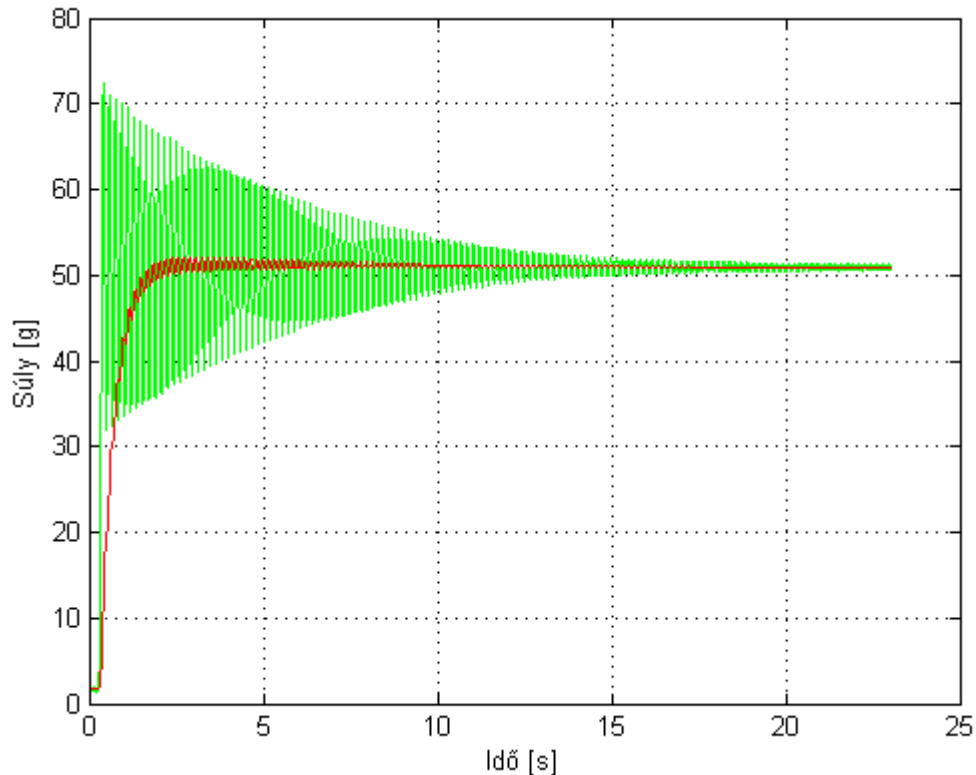
6.1 Exponenciális átlagolás  $\alpha=0.001$  és  $0.0001$ -es paraméterrel

A zölddel ábrázolt görbe az átlagolás nélküli időfüggvény, ennek az adataiból lett a többi görbe kiszámítva. A kék színű görbe  $\alpha = 0.001$  értékkel lett átlagolva. Az eredmény az lett, hogy egy rövid, közel 1 másodperces, felfutási szakasz után az átlag a végérték körül leng, majd az idő múlásával ez a lengés csillapodik, ahogyan az időfüggvény is. A pirossal ábrázolt görbe  $\alpha = 0.0001$  paraméter alkalmazásával készült.



Látható, hogy ilyen kicsi paraméter választása esetén a beállási idő nagyon megnövekedett, de a lengés szinte teljesen eltűnt.

Ezek után választottam egy olyan  $\alpha$  értéket, ami a két előző közé esik és ezzel teszteltem a módszert. Az eredmény az 6.2-es ábrán látható.



### 6.2 Exponenciális átlagolás $\alpha=0.0005$ paraméter alkalmazásával

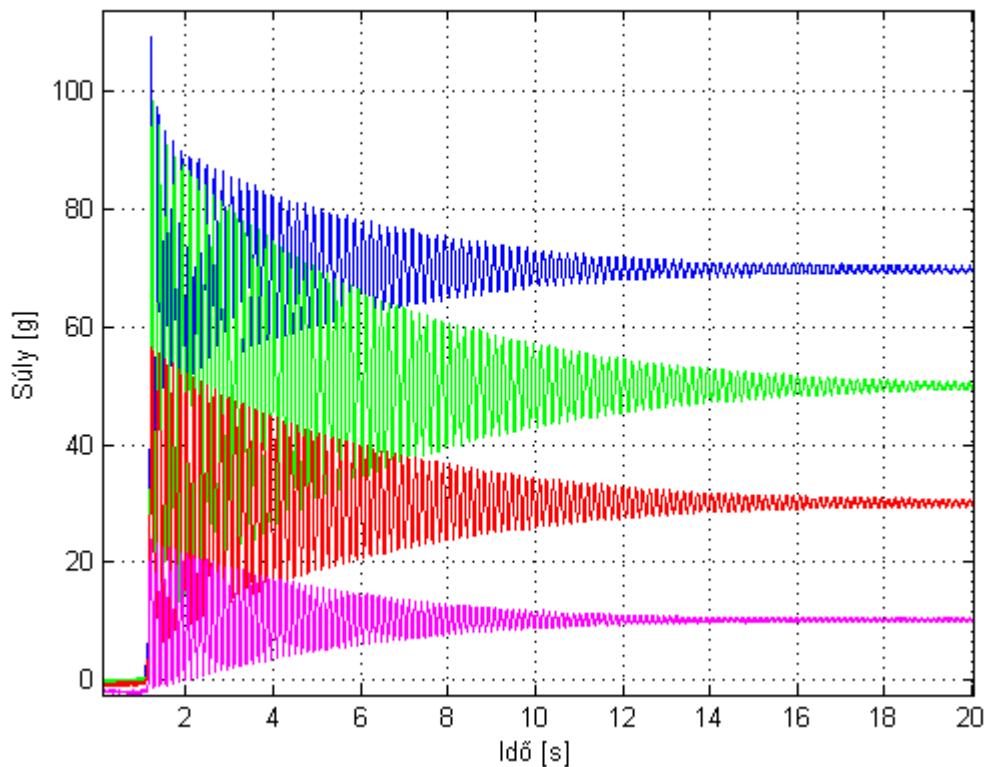
Az  $\alpha = 0.0005$  paraméterválasztás esetén ötvöződik az előző két paraméter előnye. A beállási idő kb. 2 másodperc, de szemmel látható módon az apró lengések kb. 10 másodperc elteltével tűnnek el.

A módszer előnyös tulajdonságai közé tartozik, hogy kis tárhely igénye van, nagyon gyorsan és folyamatosan számolható, hiszen csak két szorzás és egy összeadás. Teljesen érzéketlen a nulladik elem helyére és értékére és jó a zavartűrése is. A módszer egyszerűen implementálható és könnyen hangolható.

### 6.2 Kezdeti meredekség alapján számítás

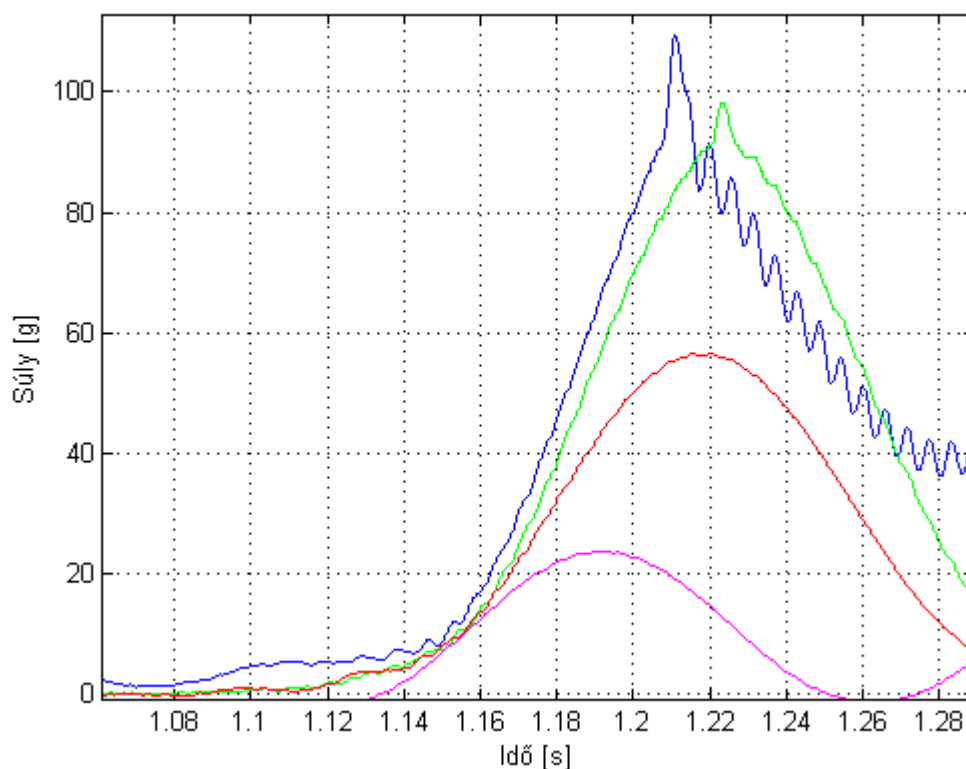
Ahhoz, hogy ezt a módszert eredményesen lehessen használni biztosítani kellett, hogy azonos kezdeti feltétele legyen a mérleg serpenyőjének minden méréskor. Ez azt

jelenti, hogy amint elindul a mérés csak a felhelyezett test tömegével legyen arányos a kezdeti kimenet külső zavaroktól mentesen. Ilyen zavar lehet, ha néhány mm magasról „esik” a serpenyőre a test, vagy a test felhelyezése előtt leng a serpenyő. Ezek elkerülésének legcélszerűbb módja, ha a nyúlásmérő bélyegeket tartalmazó lemezre kitámasztott állapotban helyezük fel a mérni kívánt tömegeket, majd a kitámasztás oldalra való gyors kihúzásával indítjuk a mérést. A különböző súlyok tesztmérésekor kapott eredményeket a 6.3-as ábrázolja:



6.3-as ábra

A rózsaszín időfüggvény 10 g-os súly mozgását ábrázolja, a piros 30 g-osat, a zöld 50 g-osat a kék pedig 70 g-osat. A 6.4-es ábra közeli kép ezek felfutásáról:



6.4-es ábra

Az ábra jól illusztrálja, hogy a megalkotott modell helyes, nagyobb tömeghez meredekebb kezdeti felfutás tartozik. Viszont jól látható több probléma is, ami megnehezíti a valós idejű feldolgozást és súlybecslést. A meredekség számítása úgy történne, hogy a felfutás két különböző értékének (lehetőleg a közel lineáris szakaszon lévőknek) különbségét el kellene osztani a köztük eltelt időkülönbséggel. A pontosabb eredmény érdekében a két mérési pontnak minél közelebb kellene lenni a felfutás két széléhez. Az első probléma, hogy nem lehet meghatározni egyértelműen a felfutás kezdetét. A felfutás végét a csúcérték 90%-nál szokták jelölni, ennek a meghatározása lehetséges az 5.4-es fejezetben ismertetett csúskereső algoritmussal majd egy 0.9-es szorzással, de látható hogy nagyobb súlyok esetén sem pontos. A harmadik nehézséget az adja, hogy a súly növekedésével a felfutási idő is nő, így amíg az utólag vizsgált adatokból látható meddig tart ez a szakasz, valós idejű feldolgozásnál nagy adathalmaz tárolása nélkül erre nincs lehetőségünk. Ekkora adatsokaságot csak a külső memóriában lehet tárolni viszont ez jelentősen lassabb, mint a belső regiszterek használata. Összességében ez a módszer nem tűnik megfelelőnek a súlybecslés gyorsítására.

## 6.3 Amplitúdók csúcsainak átlagolása

Ebben az alfejezetben az 5.5 alfejezetben ismertetett eljárások teszteredményeit és gyakorlati vonatkozásait mutatom be.

### Szélsőérték-kereső algoritmus

Az 5.5. alfejezetben bemutatott algoritmusok nagyban támaszkodnak az időfüggvény lokális minimum és maximum szélső értékeire, ezért először a szélső értékek keresésére használt algoritmust mutatom be.

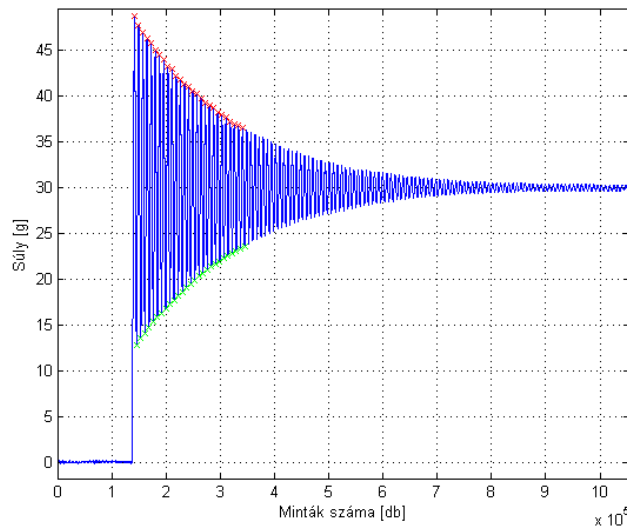
A lokális szélsőértékeket (pozitív vagy negatív csúcsok) kereső algoritmus egyesével nézi végig sorban egymás után az elemeket az elmentett mérési adatok között. Az egész a csúcsok keresésével kezdődik. A Matlab-os szimulációnál mindig valami 0 közeli értékről indult a rendszer. A csúcsok keresése akkor kezdődött, ha megérkezett a trigger jel. Ez egy hangolható paraméter volt, ami megadta hogy amint ekkora értékkel eltér a mért érték a legelsőtől, indulhat a keresés.

Mind a minimum, mind pedig a maximum értékekhez tartozik egy iterátor, amivel az algoritmus lépteti a tömbelemeket és vizsgálja az értékét, mert amint megvan a keresett számú csúcs, továbbléphet. A továbblépési feltétel, hogy a két iterátor egyenlő, és értékük pontosan a keresett csúcsok számánál kettővel több, azért kettővel mert keres 1 plusz elemet, és a tömbök indexelése miatt egyel túl kell lépnie.

Előre meg volt határozva, hogy hány darab minimum és maximum csúcs adatát kívánjuk feldolgozni. Ezek száma mindig azonos. Biztonsági és zavartűrési okokból ennél a számnál többet keres az algoritmus, mivel néha előfordult, hogy az első megtalált csúcs kisebb volt a másodiknál. Ha ilyen előfordult, az algoritmus ezt felismeri, ezt az adatot eldobja, és a többivel számol tovább. A csúcsok adatainak tárolása 4 tömbben történt. Ezekből egy-egy tárolja a maximum értéket és a hozzá tartozó időkoordinátát, a másik kettő pedig ugyanezeket az adatokat a minimum értékekre.

A csúcskeresés maximumkereséssel indul. Amint a jelenleg vizsgált elem nagyobb, mint a keresés elkezdése óta talált legnagyobb, az ő értéke felülírja az előző legnagyobb értéket. Hasonlóan a kezdeti trigger jelhez, ha a maximumhoz képest egy előre megadott nagyságú visszaesés van (ezt 3 g-ra állítottam, mivel nem voltak ekkora tüskék a zajokból adódóan), akkor az algoritmus eltávolítja a megtalált maximum értéket,

lépteti eggyel az iterátorát és a jelenlegi elem értékéről kezdi a minimumkeresést. Analóg módon a maximumkereséshez itt is, amint kisebb értéket talál, akkor felülírja az előző értékeket, majd ha az utoljára talált minimum értékhez képest 3 g-os emelkedés van, akkor lépteti a minimum iterátort, és új maximumkeresést kezd. Innen pedig változva követik egymást a minimum- és maximumkeresések, amíg az iterátorok el nem érik a leállási értéket. Az algoritmus hatékonyságát a 6.3-as szemlélteti:



### 6.3 A megtalált szélsőértékek az adatsoron

Ezután jön az esetleges hibás értékek eldobása, majd a regresszió számítás. Ez a folyamat az 5.5-ös fejezetben lett részletesen bemutatva, a Matlab kódban nincs jelentős eltérés a leírt módszertől.

### Csillapítás számolása két pontból

A módszer használhatóságának vizsgálatát az előzetes mérési eredmények feldolgozásával kezdtem. Első megközelítésben a csillapítás meghatározhatóságát teszteltem két pontra történő illesztéssel (lásd 5.5 alfejezet). A 6.1-es táblázat mutatja az eredményeket.

A táblázat első oszlopában a méréskor használt súly értéke található, a második oszlopban a felhelyezést követő első amplitúdó maximum. A 3-5. oszlopokban különböző csúcsokból számolt  $\zeta$  értékek láthatóak. A harmadik oszlopban az első 2 csúcsból, a negyedik oszlopban az első és harmadik csúcsból, az ötödik oszlopban pedig az első és a lecsengés végéhez közeli csúcsból kiszámított érték szerepel. A számításokat az (5.7) képlet alapján végeztem. Az eredmények azt mutatják, hogy a

konstansnak feltételezett  $\zeta$  többet változik egyes méréseken belül, mint ami a számolási pontatlanságokból eredhet.

Tömeg [g]	Első amplitúdó max. [g]	$\zeta_{1-2}$	$\zeta_{1-3}$	$\zeta_{1-N}$
70	91	0.2602	0.1954	0.2335
70	113	0.7914	0.6110	0.1894
70	85	0.3720	0.2809	0.1609
70	103.5	0.2364	0.2012	0.1904
30	37.25	0.4472	0.3478	0.1721
30	48.75	0.3416	0.3053	0.2579
30	41.75	0.4165	0.2790	0.1618
30	67	0.3334	0.2865	0.1727
10	20.63	0.3351	0.2526	0.2275
10	19	0.1988	0.2537	0.2774
10	16.63	0.7261	0.4729	0.2883
10	16.13	1.9919	1.1960	0.3105
50	85.5	0.2439	0.2057	0.1935
50	72.5	0.4011	0.2682	0.1754
50	62	0.2431	0.2493	0.1991

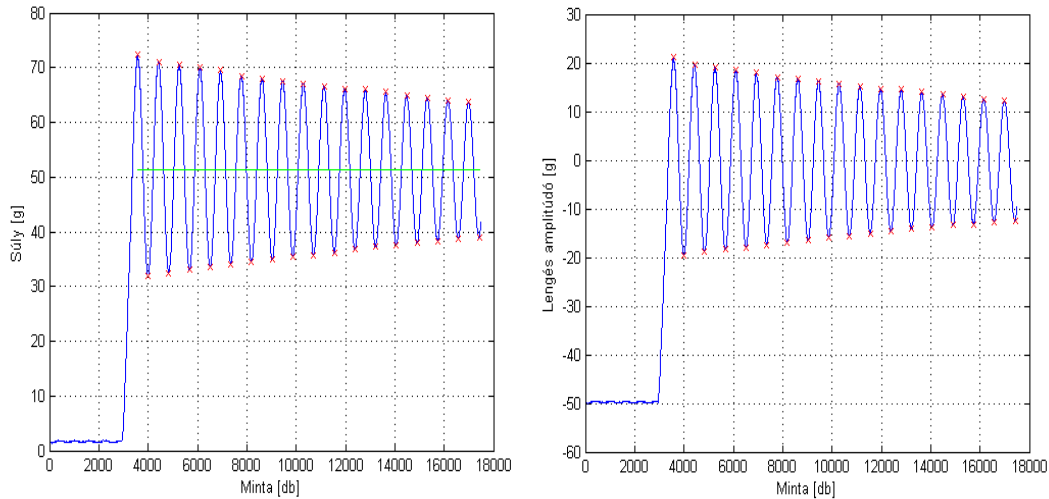
#### 6.1 Csillapítás számítása amplitúdó csúcsokból

Az eredmények alapján a két pontra történő illesztés bizonytalanságokkal terhelt, ezért érdekesebb volt több pont felhasználásával számolni.

#### Csillapítás számítása több pontból és súlybecslés

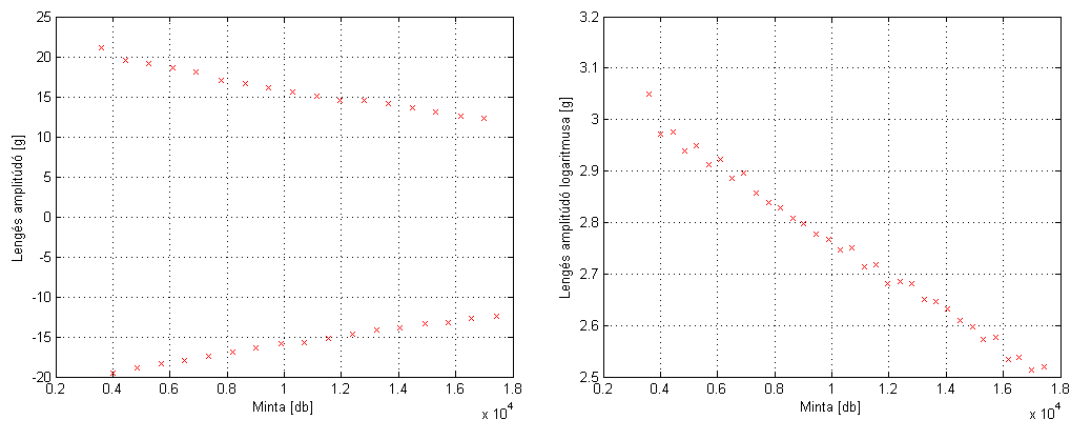
A következő szakaszban az 5.5. alfejezetben bemutatott becslési eljárások egyes lépéseit mutatom be részletesen.

Az algoritmus veszi a megtalált csúcsok ( $y_i$ ) számtani közepét ( $x_0$ ) az (5.10) egyenlet alapján. Ez a középérték a 6.4-es ábrán zöld vonallal van jelölve.  $x_0$  tekinthető egy első becslőnek a tömegre vonatkozóan, másrészt a súlyozott átlagolás elvégzéséhez is szükséges kezdőérték. Ezzel az  $x_0$  értékkel kell eltolni az időfüggvényt, hogy a középértéke a nullába kerüljön (lásd 6.4 jobb oldali ábra).



### 6.4 Az algoritmus lépései 1

Az ábrán kék színnel van ábrázolva az időfüggvény, zölddel az értékek egyszerű átlaga és pirossal a megtalált amplitúdó csúcsok. A késsel jelölt időfüggvényt nem használtam a további feldolgozáshoz, csupán illusztrációként jelenítettem meg.



### 6.5 Az algoritmus lépései 2

A 6.5 ábra bal oldalán a megtalált csúcsokat kiemelve ábrázoltam, míg a jobb oldali ábrán ezen pontok abszolút értékének logaritmusai láthatóak. Az exponenciális lecsengés  $\zeta$  paraméterét úgy kapjuk meg, hogy  $e$ -adik hatványra kell emelni a logaritmus függvényre illesztett egyenes meredekségét.  $\zeta$  felhasználásával az (5.12) és a (5.13) egyenlet alapján megkapjuk becsült súly nagyságát.

Az algoritmust az elkészítés után 25 db mérési adatsoron futtattam végig. Állítható volt a megkeresett csúcsok száma, ezért minden adatsor végig lett tesztelve 5-20 csúcscsám esetén. A teszt kiszámolta, hogy adott csúcscsám mellett mennyi az (5.10)

egyenlet alapján számított súlyozatlan átlag, az (5.13)-as képlettel számított exponenciálisan súlyozott átlag, a végérték és a két átlag %-os hibája. A kapott eredményeket a 6.2-es táblázat tartalmazza.

Csúcscsám	Átlagos eltérés(egyszerű,%)	Átlagos eltérés(exp,%)	Szükséges becsült idő (s)
5	3,55	3,05	0.875
6	3,43	2,93	1.05
7	3,32	2,84	1.225
8	3,24	2,76	1.4
9	3,16	2,69	1.575
10	3,10	2,63	1.75
11	3,03	2,57	1.925
12	2,97	2,51	2.1
13	2,91	2,45	2.275
14	2,86	2,41	2.45
15	2,81	2,36	2.625
16	2,77	2,32	2.8
17	2,73	2,28	2.975
18	2,69	2,24	3.15
19	2,65	2,20	3.325
20	2,61	2,16	3.5

**6.2 Az egyszerű és az exponenciális átlagolás hibái**

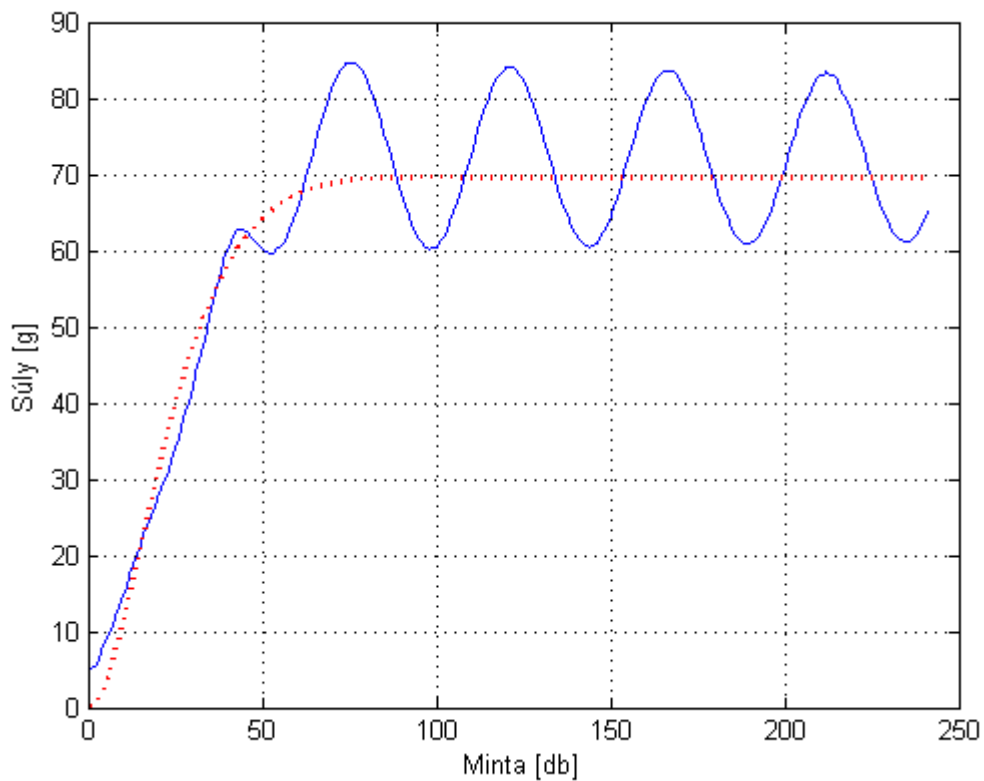
Várható volt, hogy a keresett csúcsok számának növekedésével a hibák csökkenni fognak. Minden újabb csúcspár (minimum-maximum) megkeresése 150-200 ms-ot vesz igénybe. Az 5-nél több csúcspár adatainak felhasználása elég jó becslést biztosít, de a felhasznált csúcsok számát növelve jelentősen nő a becslés időszükséglete. Ezen okokból kifolyólag a DSP-re lekódolt programban 5 csúcspárral van megvalósítva az átlagolás, mivel gyors becslés volt a cél

## 6.4 Rendszeridentifikáció

Az 5.6-os fejezetben ismertetett elmélet alapján először az  $\mathbf{U}$  mátrixot kellett összeállítani. Ennek 3 oszlopa van és az  $n$ -edik ütemben a következő elemeket tartalmazza. Az első tartalmazza az  $u(n)+2u(n-1)+u(n-2)$  értéket, ahol  $u(n)$  a gerjesztés értéke az  $n$ -edik ütemben. Mivel  $u(n)$  egzakt értéke nem ismert, gerjesztésként egységugrást feltételeztem. A második oszlopban az  $y(n-1)$  van, ez a válasz eggyel shift-elt értéke, a harmadik oszlop pedig  $y(n-2)$ .

Az (5.32)-es egyenlet alapján  $\mathbf{U}$  és  $\mathbf{y}$  birtokában meghatározható  $\mathbf{p}$ . A paraméterekből az (5.33)-as egyenlettel leírt rendszer meghatározható. Egy konkrét mérési adatsorra lefuttatott identifikáció eredménye 6.8-as ábrán látható.





**6.8 Az identifikált rendszer ugrásválasza (piros)**

A képen pirossal látható az identifikált rendszer ugrásválasza és késsel az adatsor, ami alapján az identifikáció készült. Az ugrásválasz beáll a végértékhez, de nem leng. Ennek okozója lehet, hogy ez a módszer érzékeny a megadott gerjesztésre, illetve olyan közelítéssel éltem, hogy a gerjesztés egy egységugrás. A valós gerjesztés pontosan nem ismert, csak a próbaképpen felhelyezett súly, és valószínűleg nem pontszerű esemény a felhelyezés sem. Az identifikált rendszer nagyobb fokszám esetén sem illeszkedett eléggé a mért adatokra. A méréseket ellenőrzésként a Matlab System Identification toolbox-a segítségével is elvégeztem, és jellegre hasonló eredményeket kaptam.

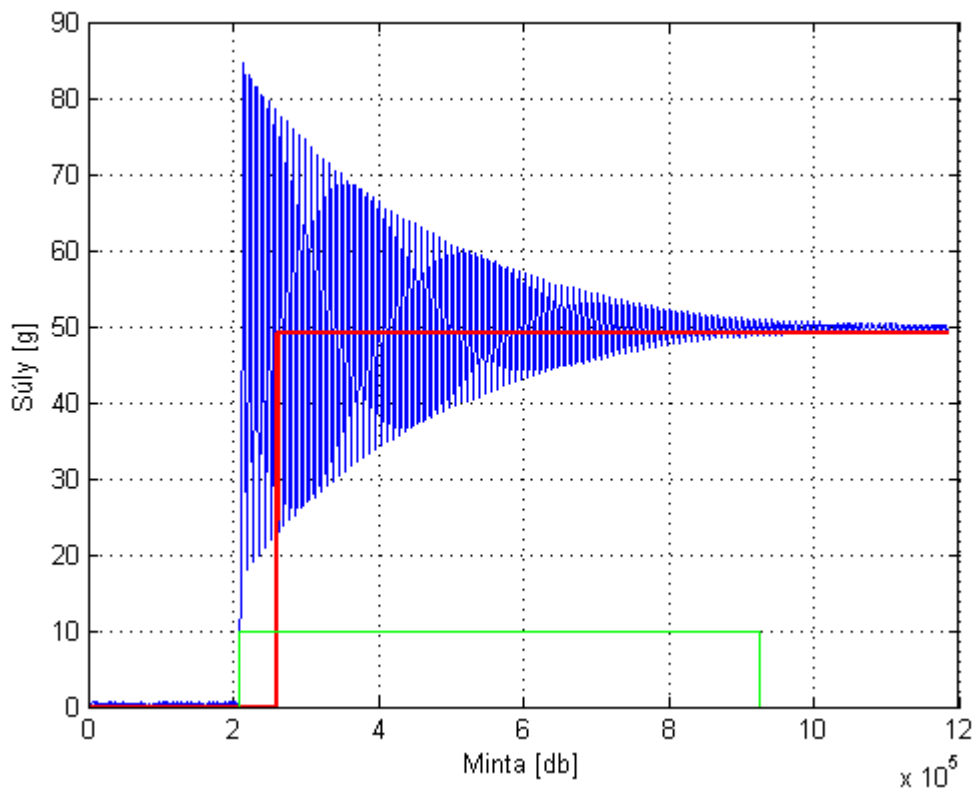
Habár a tapasztalatok alapján a módszer versenyképes lehet, a végső rendszerben történő implementálását elvettem a következő két ok miatt:

- Mivel a rendszer viselkedését nem tudtam egzaktul leírni, így további megfontolásokat igényel, hogy milyen ponton kell javítani az eljárást.
- A számítás nagyméretű mátrixok és vektorok kezelését igényli, és mátrixinverziót is tartalmaz. A valós idejű számítás és a beágyazott rendszerek szűkös erőforrásai miatt a megvalósítás tehát komoly problémákba ütközhet.

## 6.5 DSP-re írt szoftver online méréshez

Az eddigi alfejezetben bemutatott eljárások közül az amplitúdó csúcsok súlyozott átlaga volt a legígéretesebb becslési eljárás, ezért ezt implementáltam a jelfeldolgozó processzorra.

Alapjaiban megegyezik a Matlab-os offline tesztek során használt algoritmussal, de tartalmaz néhány eltérést. Az egyik legfontosabb különbség, hogy a programot fel kellett készíteni olyan esetekre is, amikor a becsülni kívánt súly kevesebb, mint a becslés elkezdésekor a serpenyőben lévő (tipikus esete, ha leveszünk egy tárgyat a serpenyőről). A másik nagyon fontos különbség, hogy periodikusan működő szoftverre volt szükség. Ez azért lényeges, mert a mérleg használata során határozatlan ideig mérünk, gyakran több különböző súlyt egymás után. Fontos tehát, hogy a becslő algoritmus észlelje az új becslés elkezdésének szükségét.

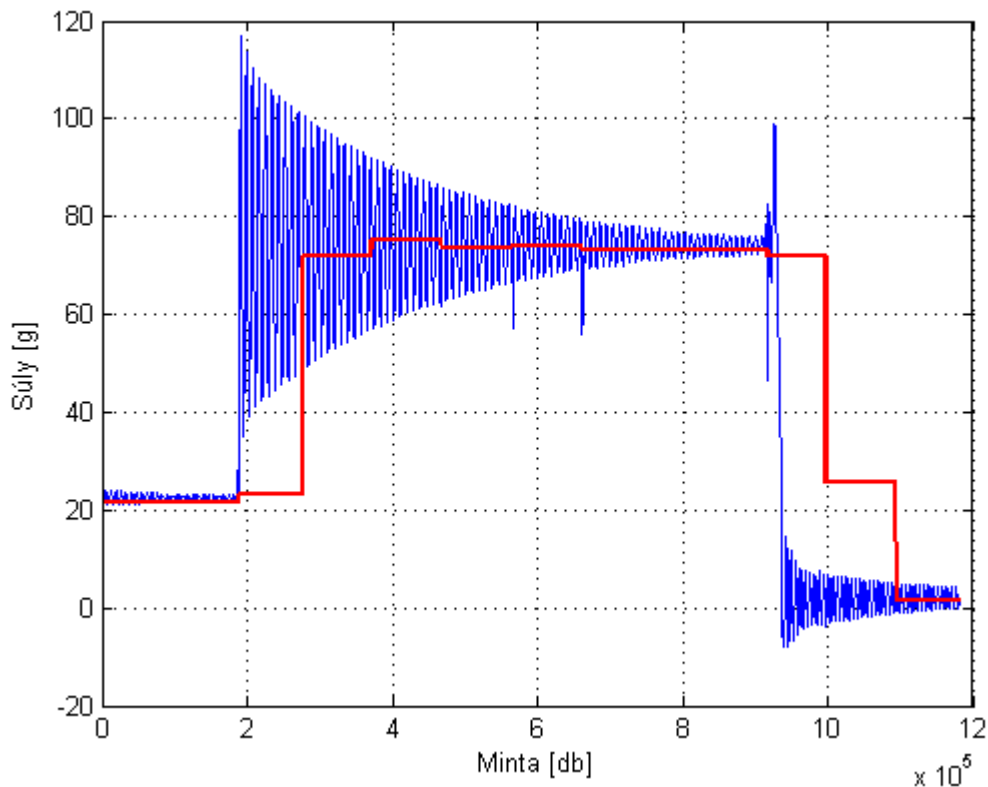


6.6 Egyszeri becslés DSP-n

A 6.6-os számú ábra egy becslést mutat be, amely online módon a DSP-n lett elvégezve. Minden adatot a DSP számolt, a Matlab-ban csak az ábrázolás lett elvégezve. Az ábrán kék színű pillanatnyi érték, zöld a becslés kezdetét jelző trigger és

piros a becsült érték. A triggerfeltétel az, hogy a korábbi becsült értéktől 4 g eltérés mérhető.

A periodikus működés úgy lett megvalósítva, hogy a program 1.5 s-nál sűrűbben nem végez becslést. Ez a paraméter módosítható. Amint letelt ez az idő, megnézi, hogy érkezett-e trigger újabb becslés elindításához, majd a válasznak megfelelően cselekszik. Az algoritmus viselkedését és az új helyzetekre való reakcióját az 6.7-es ábra szemlélteti.



### 6.7 Periodikus becslés

Az ábrán kék színű a pillanatnyi amplitúdó és piros színű a becsült érték. A bemutatott ábra alapján látható, hogy ez a módszer magában hordoz minden fontos tulajdonságot, amit egy jó becsülő módszertől elvárhatunk. Gyors, pontos, jól reagál a változásokra, elfogadható számítás igényű és folyamatosan működtethető.

## 7 Összefoglalás és kitekintés

Szakedolgozatom keretében egy mechanikus mérleg villamos jeleinek szoftveres feldolgozását valósítottam meg. Létrehoztam egy egyszerűen kezelhető tömegmérő rendszert, amelynek a méréstartománya 0-100g.

A dolgozat készítése során a fejlesztésnek két fő iránya volt: egy megjelenítő egység illesztése, valamint a mérendő súly becslésére egy gyors és pontos módszer kidolgozása, implementálása és tesztelése. A megjelenítőt tartalmazó eszköz rendelkezésre állt, így nekem az illesztő szoftvert kellett megvalósítani.

Szoftveres becsléssel, több módszer tesztelése után, sikerült lecsökkentenem a mérleg 15-20 másodperces beállási idejét 2 másodperc alá. A becsléshez az amplitúdócsúcsok átlagolásának módszerét implementáltam a DSP-re, mivel ez pontosabb és gyorsabb a többi módszernél, valamint majdnem annyira robusztus, mint az exponenciális átlagolás. A megjelenített érték mindig a legutolsó becslés eredménye.

A jövőben a mikrokontroller szoftverét érdemes tovább fejleszteni, a chipet tartalmazó nyákon lehetőség van nyomógombok és forgatógombok használatára. Ezekkel online módon lehetne állítani a kalibrációs pontokat, a becslés során használt paramétereket (pl.: csúcspárok száma, visszaesés nagyság, új becslés indítása előtti várakozás hosszúsága). Ehhez szükség lenne egy menüre és kétirányú kommunikációs csatornára a DSP felé.

Érdemes lenne még a becslési algoritmusokat is tovább fejleszteni. Az identifikáció ígéretes eredményeket mutatott, lehetne foglalkozni azzal, hogy jobban illeszkedjen az időfüggvényekre, illetve a jelfeldolgozó processzorra kellene optimalizálni az algoritmus számítási lépéseit. Tesztelni lehetne a bemutatott módszerek ötvözését is, hátha ki lehet nyerni az előnyös tulajdonságaikat, a hátrányaik nélkül.

## Irodalomjegyzék

- [1] Mérlegek: [http://hu.wikipedia.org/wiki/Mérleg\\_\(eszköz\)](http://hu.wikipedia.org/wiki/Mérleg_(eszköz))
- [2] ADSP-BF534\_BF536\_BF537.pdf, <http://analog.com>
- [3] Hudson, A. –Nelson, R. , *Útban a modern fizikához*
- [4] <http://www.edtc.com>
- [5] ATmega32 datasheet, <http://atmel.com>
- [6] <http://mathworks.com>
- [7] <http://hu.wikipedia.org/wiki/Rezgés>
- [8] BME MIT tanszéki közösség, *Digitális jelfeldolgozás jegyzet*
- [9] Lantos Béla, *Irányítási rendszerek elmélete és tervezés I.*