

Ismo Kemppainen

**Digitaalisen suodattimen kokeilu LabVIEW-ohjelmointiympäristössä**

Insinöörityö  
Kajaanin ammattikorkeakoulu  
Tekniikan ja liikenteen ala  
Tietotekniikan koulutusohjelma  
Syksy 2006



**Kajaanin  
ammattikorkeakoulu**

## OPINNÄYTETYÖ TIIVISTELMÄ

Koulutusala Tekniikka	Koulutusohjelma Tietotekniikka
Tekijä(t) Ismo Kempainen	
Työn nimi Digitaalisen suodattimen LabVIEW-ohjelmointiympäristössä	
Vaihtoehtoiset ammattiopinnot Elektroniikan testaus	Ohjaaja(t) Jukka Heino Toimeksiantaja Jukka Heino
Aika Syksy 2006	Sivumäärä ja liitteet 28+2
<p>Insinööri työn tavoitteena oli tutkia, kuinka tehokkaasti LabVIEW-ohjelmointiympäristö toimii muunnettaessa signaali taajuustason signaaliksi. Työn tavoitteena oli myös tutkia, kuinka pitkän nopean Fourier-muunnoksen LabVIEW-ohjelmointiympäristö kykenee laskemaan virheettömästi. Signaali välitetään ohjelmaan PC-tietokoneen äänikortin A/D-muuntimen kautta. Ohjelman oli myös tarkoitus välittää signaali PC-tietokoneen äänikortin D/A-muuntimen kautta.</p> <p>LabVIEW-ohjelmointiympäristössä on valmiit suodatinfunktiot, mutta tässä työssä rakennettiin oma suodatin. Ohjelma muuntaa aikataason signaalin taajuustason signaaliksi. Signaalinkäsittely on taajuustasossa helpompi toteuttaa verrattuna aikataason signaaliin.</p> <p>Tässä työssä saavutettiin kaikki sille asetetut tavoitteet. Suodatin ei toiminut täysin koska LabVIEW ei pitänyt sisällään amplitudispektrille tarkoitettua IFFT-funktiota. Voidaan sanoa, että tämän tyyppistä signaalin suodatinta ei voida toteuttaa LabVIEW-ohjelmointiympäristössä.</p>	
Kieli	Suomi
Asiasanat	FFT, Virtuaali Instrumentti, Filter, A/D- and D/A -muunnin
Säilytyspaikka	<input type="checkbox"/> Kajaanin ammattikorkeakoulun Kaktus-tietokanta <input type="checkbox"/> Kajaanin ammattikorkeakoulun kirjasto

School Engineering	Degree Programme Information Technology
Author(s) Ismo Kemppainen	
Title Testing of a Digital Filter in the LabVIEW Programming Language	
Optional Professional Studies Electronic Testing	Instructor(s) Jukka Heino
	Commissioned by Jukka Heino
Date Autumn 2006	Total Number of Pages and Appendices 28+2
<p>The goal of this Bachelor's thesis was to examine the signal processing capability of the LabVIEW programming language when the signal is converted to the frequency domain signal. The objective of the thesis was also to examine the FFT size that an application can calculate without faults. The signal was transferred to the application through the A/D converter of the PC sound card. The objective of the application was also to transmit the signal out of the PC through the D/A converter of the PC sound card.</p> <p>The LabVIEW programming language has its own filters but the objective of this thesis was to build a filter. The application converts the first time domain signal to the frequency domain signal. Signal processing was easier to realize in the frequency domain signal than in the time domain signal.</p> <p>In the thesis all the requirements that were set on it were met. The filter did not work because LabVIEW does not contain the power spectrum IFFT Virtual Instrument. It can be said that this kind of signal filter cannot be realized with the LabVIEW programming language.</p>	
Language of Thesis	Finnish
Keywords	FFT, Virtual Instrument, Filter, A/D and D/A converter
Deposited at	<input type="checkbox"/> Kaktus Database at Kajaani University of Applied Sciences <input type="checkbox"/> Library of Kajaani University of Applied Sciences

## ALKUSANAT

Tämä insinöörityö on tehty Kajaanin ammattikorkeakoululle langattoman tiedonsiirron tarpeisiin. Työn valvojana ja ohjaajana toimi Jukka Heino, ja hän on myös työn toimeksiantaja. Haluan kiittää häntä työn ohjaamisesta.

Haluan kiittää myös kielellisessä ohjauksessa avustanutta Eero Soinista sekä englannin kielen abstraktin ohjauksessa avustanutta Kaisu Korhosta.

Kajaanissa 28.11.2006

Ismo Kemppainen

## SISÄLLYS

1 JOHDANTO	2
2 SIGNAALI	3
3 NOPEA FOURIER-MUUNNOS	4
3.1 Laplace-muunnos	4
3.2 Fourier-sarjat	5
3.3 Fourier-muunnos	6
3.4 Cooley-Tukey-algoritmi	7
3.5 Prime-Factor-algoritmi	7
4 MATLAB	8
5 SUODATTIMET	10
5.1 Suodatintyypit	10
5.2 Digitaalinen signaalinkäsittely	11
5.3 Digitaalisten signaalinsuodattimien tyypit	13
6 LABVIEW	16
7 OHJELMAN TOTEUTTAMINEN	18
7.1 Signaalin A/D-muunnos	18
7.2 Suodattimen toteuttaminen	20
8 TESTAAMINEN	24
9 TYÖN TULOKSET	25
10 YHTEENVETO	26
LÄHTEET	27
LIITTEET	

## 1 JOHDANTO

LabVIEW on mittaustekniikan puolelle suunniteltu ohjelmankehitysympäristö, jonka avulla on mahdollista toteuttaa oikeita mittalaitteita muistuttavia käyttöliittymiä. Tämän insinöörityön tarkoituksena oli selvittää voiko LabVIEW:n valmiilla VI-funktioilla toteuttaa ohjelmallisen signaalin suodattimen ja kuinka pitkän FFT-muunnoksen LabVIEW-ohjelmointiympäristöllä toteutettu suodatin kykenee laskemaan virheettömästi. Selvitystä tarvittiin langattoman tiedonsiirron tarpeisiin Kajaanin ammattikorkeakoululle.

Työssä herätesignaali siirretään ohjelmointiympäristöön PC-tietokoneen äänikortin A/D-muuntimen avulla. Aikatason signaali muunnetaan taajuustasoon FFT-muunnoksella, jossa signaalin suodattaminen tapahtuu. Tämän jälkeen signaali muunnetaan IFFT-muunnoksella takaisin aikatason signaaliksi. Näin signaali voidaan muuntaa takaisin analogiseksi PC:n äänikortin D/A-muunninta käyttäen. LabVIEW:n omia suodatinfunktioita ei käytetty, ja näin ollen samalla syvennyttiin ohjelmalliseen signaalinsuodattimen toteuttamiseen.

## 2 SIGNAALI

Digitaalinen signaalinkäsittely on tapa käsitellä signaalia digitaalisesti. Termi digitaalinen tulee sanasta digit tarkoittaen lukumäärää tai numeerista tietoa. Laskettaessa sormilla ne voidaan ajatella yksittäisinä numeroina, ja näin ne ovat jokaisen oma numeerinen tieto. Digitaalinen signaali sisältää numeroiden jonon, yleensä binaarisessa muodossa, muttei välttämättä. [1.]

Signaalinkäsittelyssä signaalilla tarkoitetaan mitä tahansa muuttujaa, joka sisältää jonkinlaista informaatiota. Digitaalisessa signaalinkäsittelyssä informaatio on muunnettu numeeriseen muotoon.

Monissa tapauksissa haluttu signaali on mallinnettu analogisella jännitteellä tai virralla, joka on tuotettu mikrofoniin tai jollakin muunlaisella muuntimella. Joissakin tapauksissa signaali on jo valmiiksi digitaalisessa muodossa, josta hyvä esimerkki on CD-levy. Analoginen signaali on muunnettava digitaaliseen muotoon, ennen kuin digitaalinen signaalinkäsittely voidaan toteuttaa. Esimerkiksi analoginen jännitesignaali voidaan digitoida käyttäen A/D-muunninpiiriä. A/D-muunnin antaa vasteeksi binaariset numerot, jotka esittävät herätteenä olevan jännitteen tasot binaarisina. [1.]

Signaalia tarvitsee yleensä prosessoida monista eri syistä. Esimerkiksi välitettävä signaali voi sisältää ei-haluttua kohinaa. Suodattamalla signaalista saadaan poistettua tai vähennettyä ei-halutut osat. [1.]

### 3 NOPEA FOURIER-MUUNNOS

Signaalia voidaan tarkastella kahdessa eri tasossa, aikatasossa ja taajuustasossa. Aikatason esityksessä signaali esitetään ajan suhteen. Sinisignaali voidaan esittää:

$$x(t) = A \sin(\omega t + \phi),$$

jossa  $|A|$  = amplitudi,  $\omega$  = kulmataajuus ja  $\phi$  = vaihekulma. Aikatason signaali mitataan oskilloskoopilla, jossa y-akselilla ilmaistaan signaalin amplitudi ja x-akselilla aika. Taajuustasossa signaalia tutkitaan spektrianalysaattorilla, jossa y-akselilla muuttujana on signaalin amplitudi ja x-akselilla taajuus. Kaikki signaalit voidaan esittää edellä mainituissa tasoissa. Riippuen siitä, mitä signaalille halutaan tehdä, valitaan haluttu taso. Taajuustasossa signaalille tehtävät matemaattiset toimenpiteet ovat helpompia. [2.]

#### 3.1 Laplace-muunnos

Laplace-muunnos on tärkeä analysointityökalu tutkittaessa jatkuva-aikaisia analogisia signaaleja. Kun diskreettiaikainen signaali on saatu näytteistettyä, sille voidaan suorittaa Laplace-muunnos. Linkki jatkuva-aikaisen ja diskreetin signaalin välillä on hyvin pysyvä käyttäen Laplace-muunnosteoreemaa, joka voidaan esittää:

$$x(t) \stackrel{L}{\Leftrightarrow} X[s],$$

josta seuraa, että:

$$x(t - kT_s) \stackrel{L}{\Leftrightarrow} e^{-skT_s} X(s).$$

Kun aikatason signaali voidaan esittää näytteistettynä



$$x[k] \leftrightarrow \{x[0], x[1], x[2], \dots\},$$

seuraa, että:

$$x[k] = x[0]\delta[k] + x[1]\delta[k-1] + x[2]\delta[k-2] + \dots = \sum_{n=0}^{\infty} x[n]\delta[k-n].$$

Laplace-muunnos edelliselle näytteistetylle signaalille saadaan:

$$X(s) = x[0] + x[1]e^{-sT_s} + x[2]e^{-2sT_s} + \dots = \sum_{n=0}^{\infty} x[n]e^{-nsT_s}.$$

Ongelma esityksessä on se, että jokainen laskutoimitus liittyy viiveoperaattoriin  $e^{-nsT_s}$  mukaan, joka tekee laskutoimituksesta hitaan. [3.] Laplace-muunnoksen jatkokehittelmä on Fourier-sarjat.

### 3.2 Fourier-sarjat

Signaalin matemaattinen esittäminen perustuu Fourier-sarjoihin, jotka ovat lineaarinen summa harmonisesti vertailukelpoisista sinimuotoisista signaaleista. Ranskalainen matemaatikko Jean Baptiste Joseph Fourier, (1768–1830) käytti trigonometrisia sarjoja kuvatakseen lämmön johtumista ja lämpötilan jakautumista kehossa. Fourier-sarja voidaan esittää trigonometristen sin- ja cos-funktioiden summana: [4.]

$$x(t) = a_0 + \sum_{k=1}^{\infty} (a_k \cos 2\pi k F_0 t - b_k \sin 2\pi k F_0 t).$$

### 3.3 Fourier-muunnos

Laplace- ja Fourier -muunnokset ovat perustapoja jatkuva-aikaisten signaalien tutkimisessa. Diskreetti-aikainen versio jatkuva-aikaisesta Fourier-muunnoksesta voidaan esittää analyttisellä kaavalla:

$$X(e^{j\omega}) = \int_{-\infty}^{\infty} x(t)e^{-j\omega t} dt$$

ja synteettisellä kaavalla:

$$x(t) = \frac{1}{2\pi} \int_{-\infty}^{\infty} X(e^{j\omega})e^{j\omega t} d\omega,$$

jossa  $\omega$  on signaalin kulmataajuus radiaaneina per sekunti ja  $X(e^{j\omega})$  signaalin  $x(t)$  spektri. Fourier-muunnos on toimiva vain, jos  $x(t)$  on absoluuttisesti integroitavissa. Fourier-muunnos on erittäin tärkeä työkalu etsittäessä systeemien heikkouksia tietokoneaikana. [3.] Fourier-muunnoksesta on tehty monenlaisia sovelluksia nopean Fourier-muunnoksen kehittämiseksi, ja niistä yleisin on Cooley-Tukey-algoritmi.

### 3.4 Cooley-Tukey-algoritmi

Cooley-Tukey-algoritmi on eniten käytetty FFT-algoritmi. Se on nimetty J.W. Cooleyn ja John Tukeyn mukaan.

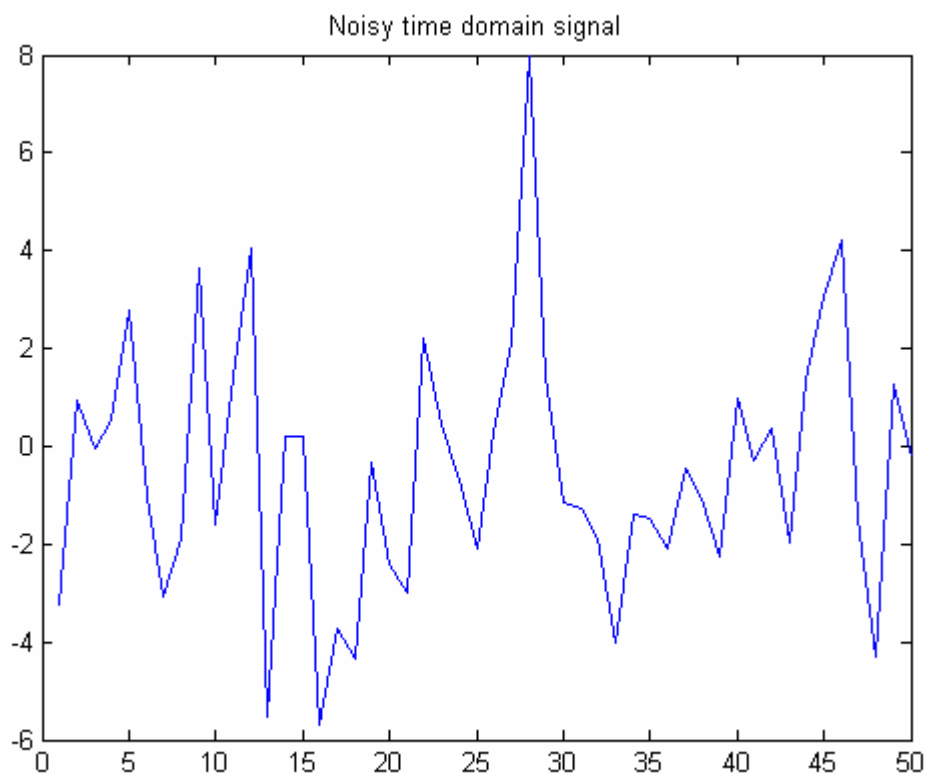
Cooley-Tukeyn algoritmin mukaan näytteiden lukumäärän ollessa kakkosen potenssi ( $N = 2^m$ ) voidaan muunnos jakaa  $m$ -vaiheiseksi sarjaksi kahden luvun Fourier-muunnoksia. Jos näytteiden määrä ei ole kakkosen potenssi, voidaan sarjan perään lisätä nollia ja sillä ei ole mainittavaa merkitystä spektriin. Algoritmilla saavutetaan merkittävä säästö laskenta-ajassa. 400-pisteisen diskreetin Fourier-muunnoksen (DFT) laskennassa tarvitaan noin 160 000 ( $400^2$ ) kompleksista kerto- ja summalaskua. Cooley-Tukey-algoritmin avulla data täydennetään noin 512 ( $2^9$ ) näytteeksi lisäämällä nollia. Näin ollen algoritmin tarvitsee laskea noin 9216 ( $2 * 512 * 9$ ) kompleksista kerto- ja summalaskua. Ajassa säästetään 17-kertaisesti. Algoritmin avulla on mahdollista toteuttaa lähes reaaliaikainen FFT-analysaattori. [5.]

### 3.5 Prime-Factor-algoritmi

Prime-Factor-algoritmi, jota myös kutsutaan Good-Thomas-algoritmiksi, on toinen tyypillinen tapa laskea DFT nopeammin. Algoritmi järjesteele DFT:n uudelleen siten, että  $N$ :n pituinen DFT muunnetaan 2-ulotteiseksi,  $N_1$  ja  $N_2$ :ksi tapauksissa, joissa  $N_1$  ja  $N_2$  ovat tarpeeksi hyvänlaatuisia. Nämä pienimmät FFT:n koon muunnokset arvioidaan uudelleen käyttäen Prime-Factorin toistettavuutta tai jotakin toista FFT-algoritmia. [3.]

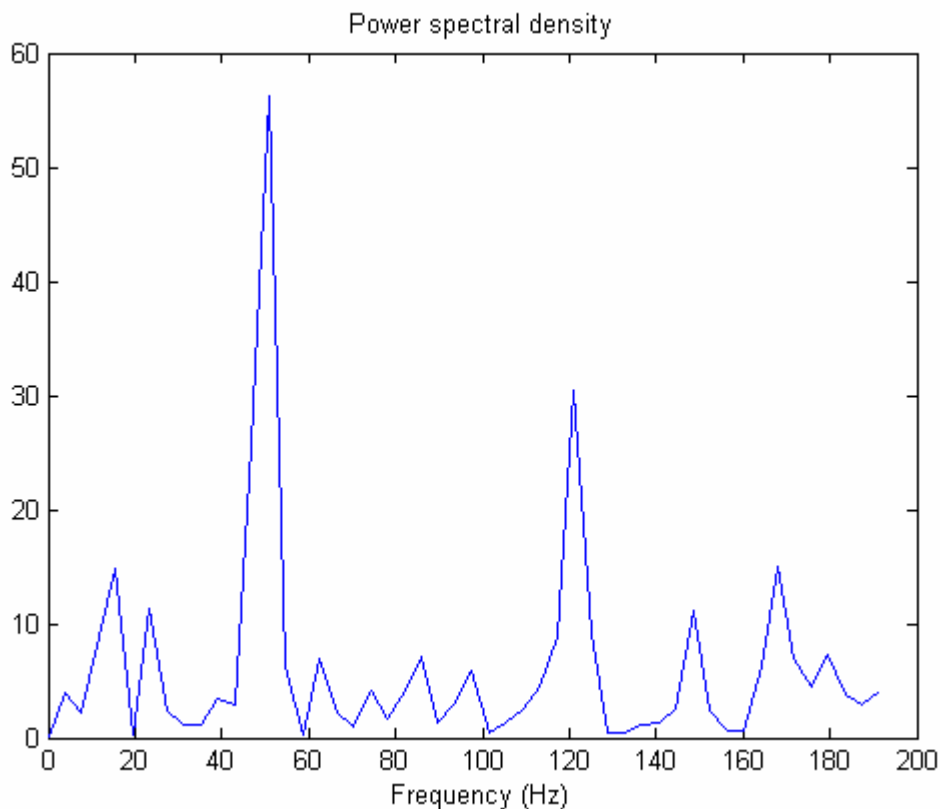
#### 4 MATLAB

Edellä esitetyn teorian tueksi mallinnetaan MATLAB-ohjelman tapaa laskea aikatasoiselle signaalille nopea Fourier-muunnos. Esimerkkiohjelmassa on moduloitu signaali, joka sisältää satunnaista kohinaa ja kaksi eri siniaallon taajuutta. Kuvaajasta havaitaan, ettei signaalin sisältämiä taajuuksia voida erottaa aikatasossa (kuva 1). Tämän vuoksi spektrianalyysit ovat tarpeellisia. Etsittäessä signaalista taajuuksia sille suoritetaan FFT-muunnos.



*Kuva 1. Aikatason signaali.*

Aikatason signaalista otetussa FFT-muunnoksessa (kuva 2) signaalin sisältämät sinisignaalin 50 Hz:n ja 120 Hz:n signaalit ovat selvästi havaittavissa spektripiikkeinä.



Kuva 2. 50 Hz:n ja 120 Hz:n taajuuksilla sijaitsevat siniaallon spektripiikit.

MATLAB laskee amplitudispektrin (liite 1) mittaamalla signaalin tehoa käyttäen kompleksista konjugaatiota. MATLAB ottaa kuvaajaan huomioon ainoastaan taulukon alkupään arvot taulukon puoleen väliin saakka. Laskettaessa 256-mittaista FFT:tä laskuissa otetaan huomioon 127 ensimmäistä taulukon arvoa, joten peilautuva osa jätetään pois.

Ohjelma laskee N-pituisen FFT:n, ja x-vektorin sisältäessä vähemmän kuin N-pituisen määrän näytteitä taulukkoon lisätään nollia. Päinvastaisessa tilanteessa taulukon pituutta lyhennetään. MATLABin käyttämä FFT-algoritmi:

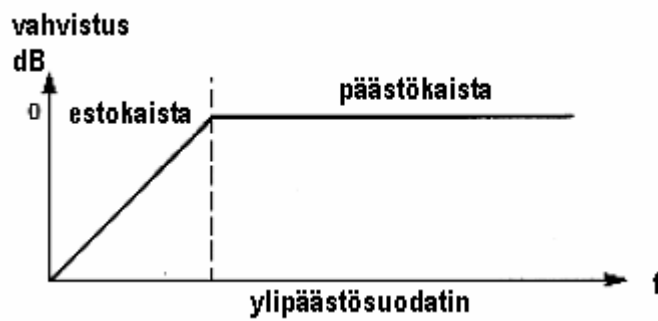
$$X(k) = \sum_{n=1}^N x(n) e^{-j*2\pi*(k-1)*(n-1)/N} .$$

## 5 SUODATTIMET

Signaalin suodattamisen tarkoituksena on poistaa signaalista ei-halutut osat, kuten satunnainen kohina, tai poimia käytettävät osat signaalista.

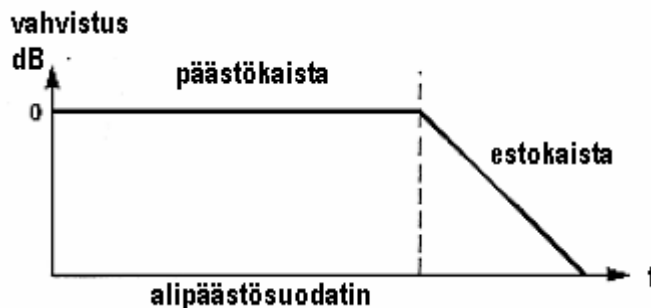
### 5.1 Suodatintyypit

Suodattimet jaotellaan kolmeen eri tyyppiin, alipäästö-, ylipäästö- ja kaistanpäästösuodattimiin. Ylipäästösuodattimella (kuva 3) korkeat taajuuskomponentit päästetään suodattimen läpi ja estokaistalla ovat rajataajuutta alhaisemmat taajuudet.



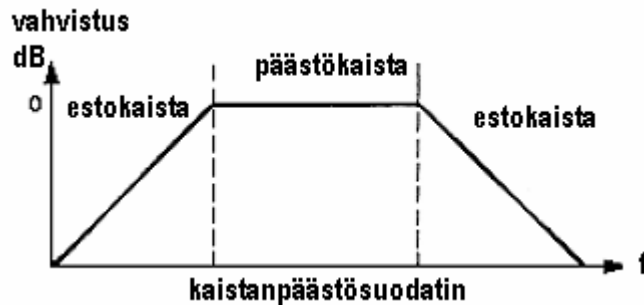
Kuva 3. Ylipäästösuodattimen esto- ja päästökaista.

Alipäästösuodattimella (kuva 4) matalat taajuuskomponentit päästetään suodattimen läpi ja estokaistalla ovat rajataajuutta korkeammat taajuuskomponentit.



Kuva 4. Alipäästösuodattimen esto- ja päästökaista.

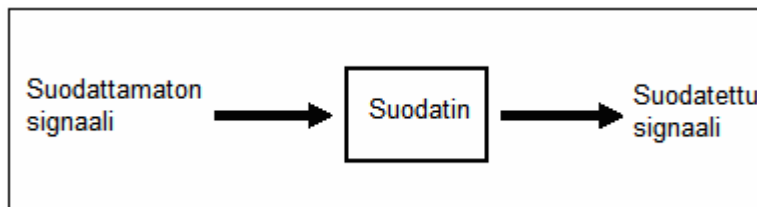
Kaistanpäästösudattimella (kuva 5) tietyt taajuuskomponentit päästetään suodattimen läpi. Estokaistoilla ovat alhaiset ja korkeat taajuuskomponentit.



Kuva 5. Kaistanpäästösudattimen päästökaista ja estokaistat.

## 5.2 Digitaalinen signaalinkäsittely

Kuvassa 6 on nähtävissä toiminnasta pelkistetty lohkokaavio.

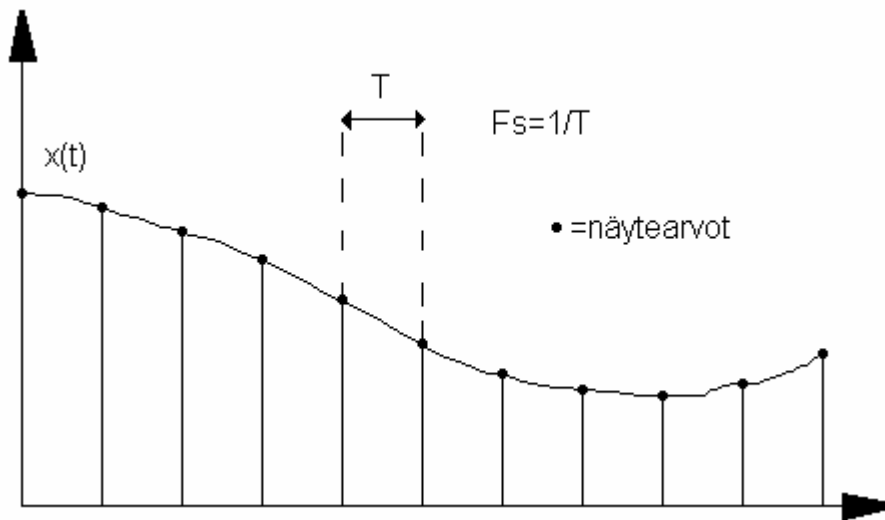


Kuva 6. Suodattimen pelkistetty lohkokaavio.

Digitaalinen signaalinsuodatin käyttää prosessoria laskeakseen numeeriset arvot signaalista otetuille näytteille. Prosessorina voidaan käyttää tavallista PC-tietokonetta tai DSP-prosessoria, joka on suunniteltu signaalin prosessointiin.

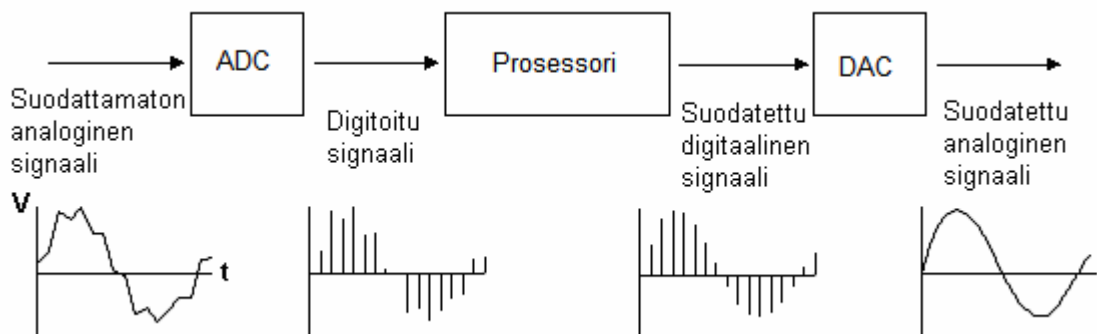
Analoginen signaali näytteistetään käyttäen A/D-muunninta. Näytteenotossa jatkuva-aikainen signaali luetaan tietyin väliajoin. Näytteenottoväli on vakio kutakin signaalia digitoitaessa. Näytteenottoväli ilmoitetaan yleensä merkinnällä  $\Delta t$ . Käänteisarvo  $1/\Delta t$  on nimeltään näytteenottotaajuus, josta merkintänä käytetään symbolia  $F_s$ . Näytteenotossa on tärkeää pitää yllä tarpeeksi suuri näytteenottotaajuus, jotta alkuperäisen signaalin sisältämä infor-

maatio ei katoa. Näytteenottotaajuutta määriteltäessä on tiedettävä suurin hetkellinen alkuperäisen signaalin taajuus. Nyquistin näytteenotto teoreeman mukaan näytteiden määrän on oltava vähintään kaksi kertaa näytteistettävää signaalia suurempi, ettei alkuperäisen signaalin informaatiota menetetä. Kun signaalin taajuus on suurempi kuin Nyquistin taajuus/2, tapahtuu laskostuminen. Tällöin alkuperäisen signaalin taajuus vääristyy ja sen spektripiikki näkyy alemmilla taajuuksilla.



Kuva 7. Näytteenotto jatkuva-aikaisesta signaalista.

Vasteena saatava perättäinen binaarinen tieto herätesignaalista välitetään prosessorille, joka laskee näytteille numeerisen arvon. Arvot tyypillisesti kerrotaan jollakin muuttumattomalla arvolla. Tämän jälkeen signaali muunnetaan takaisin analogiseen muotoon käyttämällä D/A-muunninta (kuva 8).



Kuva 8. Suodatuslaitteiston lohkokkaavio.



Digitaalisen signaalinsuodattimen ominaisuuksia voidaan muuttaa päivittämällä ohjelmistoa. Analogiseen suodattimeen verrattuna lämpötilan vaihtelut eivät muuta suodattimen ominaisuuksia.

### 5.3 Digitaalisten signaalinsuodattimien tyypit

Digitaalisia signaalinsuodattimia on perustyybiltään kahdenlaisia.

#### Konvoluutiosuodatin

Konvoluutio on matemaattinen tapa käsitellä signaaleja. Se on yleisin tekniikka signaalin prosessoinnissa. Käyttäen signaalin uudelleenjärjestelystrategiaa, systeemi on saanut aikaan signaalin, jota kutsutaan impulssivasteeksi. [6.]

Konvoluution summa voidaan esittää matemaattisesti

$$y[n] = \sum_{k=-\infty}^{\infty} (x[k]h[n-k]).$$

Jatkuva-aikaisessa signaalissa, konvoluutiota osoittaa merkki \*

$$y[n] = x[n] * h[n],$$

jossa  $y[n]$  on impulssivaste, joka on kahden signaalin,  $x[n]$  ja  $y[n]$  konvoluutio. [3.]

Tekemällä pieni muutos muuttujaan konvoluution summassa  $k = n - k$ , voidaan helposti osoittaa että konvoluutio on vaihdannainen, toisin sanoen sen suoritusjärjestyksellä ei ole väliä,

$$x[n] * h[n] = h[n] * x[n].$$

Aikatason konvoluutio on taajuustason kertolasku, joten suorittamalla aikataason signaalille FFT-muunnos, signaalille tehtävien laskutoimitusten suorittaminen helpottuu. [3.] Herä-

tesignaalin FFT:lle ja suodatinfunktion FFT:lle voidaan suorittaa kertolasku, joka tässä tapauksessa oli rectangle-ikkuna.

Taajuustason suodatin pohjautuu FFT-muunnokseen. Aikatason signaalia merkitään symbolilla  $x(t)$ . Taajuustasoon muunnetusta aikatasoon signaalista käytetään symbolia  $X(t)$ . Taajuustason suodatin voidaan esittää matemaattisesti seuraavasti:

$$Y(t) = X(t)H(t),$$

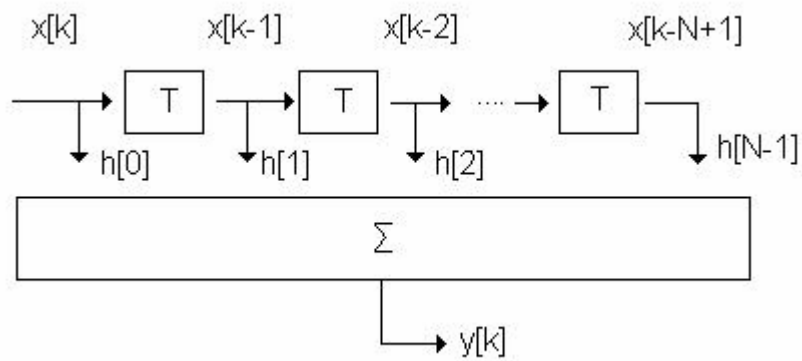
jossa  $X(t)$  on herätesignaali taajuustasossa,  $H(t)$  suodatinfunktio ja  $Y(t)$  suodatettu vastesignaali. Suodatuksen jälkeen signaali palautetaan takaisin aikatasoon IFFT-muunnosta käyttäen.

FIR- ja IIR-suodatin

FIR-suodattimella (Finite Impulse Response) on äärellinen impulssivaste, joka tarkoittaa, että suodatin tarvitsee päättävän määrän näytteitä. Impulssivaste N-kertaluvun FIR-suodattimelle voidaan esittää seuraavasti:

$$h[k] = \{h[0], h[1], \dots, h[n-1]\}$$

Se voidaan esittää kuvainnollisesti (kuva 9), josta havaitaan, että jos  $x[k] = \delta[k]$ , lähdössä  $y[k]$  on ensin  $h[0]\delta[0] = h[0]$ ,  $h[1]\delta[1-1] = h[1]$  ja niin edelleen. [3.]



Kuva 9. FIR-suodatin matemaattisesti.

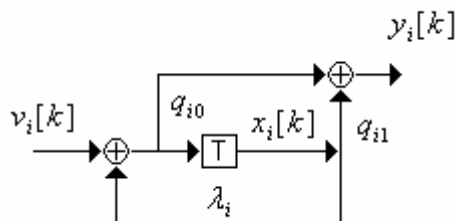
FIR-suodattimen aikataason mielivaltainen heräte  $x[k]$  voidaan kuvata lineaarisena konvoluution summana:

$$y[k] = \sum_{m=0}^{N-1} h[m]x[k-m].$$

Vaste  $y[k]$  ei pidä sisällään muuta kuin siirtorekisterin, jonka pituus on  $N-1$ , kertojan  $N$  ja akun. [3.] IIR-suodattimessa (Infinite Impulse Response) impulssivaste voi olla äärettömän pitkä vaimeneva värähtely, joka FIR-suodattimessa on äärellisen pituinen. IIR-suodatin voidaan esittää:

$$H_i(z) = \frac{q_{i0} + q_i z^{-1}}{1 - \lambda_i z^{-1}} = q_{i0} + \frac{r_i z^{-1}}{1 - \lambda_i z^{-1}},$$

jossa  $q_{i0}$  ja  $r_i$  ovat reaalikertoimia. IIR-suodattimelle (kuva 10) voidaan esittää vastaavanlainen lohkokkaaviona, kuten FIR-suodattimen kohdalla. [3.]

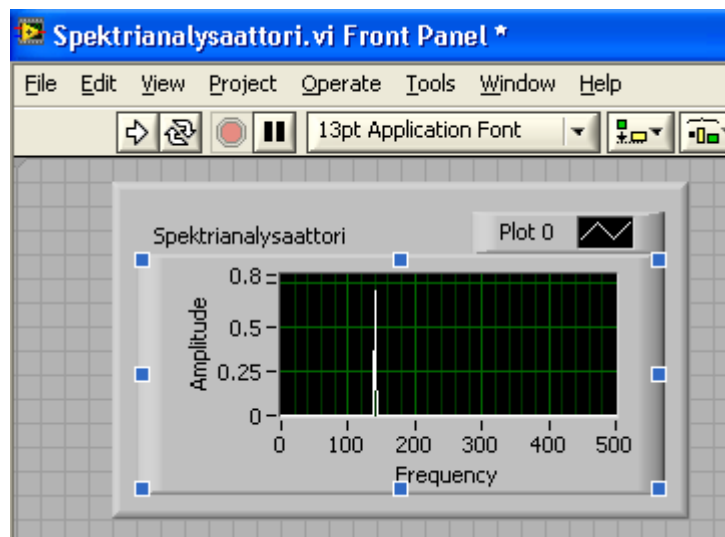


Kuva 10. IIR-suodatin.

## 6 LABVIEW

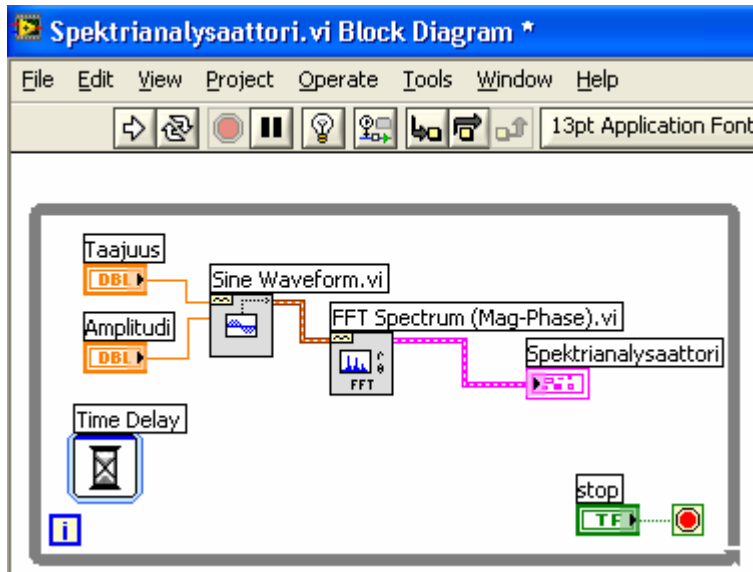
LabVIEW (Laboratory Virtual Instrument Engineerin Workbench) on ohjelmiston kehitysympäristö, joka eroaa täysin tyypillisistä tekstiohjelmointiympäristöistä kuten C/C++. LabVIEW perustuu graafiseen G-kieleen. LabVIEW on lähes de-facto standardi mittaustekniikan sovelluksissa. Sen alkuperäinen käyttötarkoituksin on suunniteltu juuri mittaustietojen keruuseen ja analysointiin. Kuitenkin ohjelman monipuolisuuden vuoksi, sitä voi käyttää myös yleisohjelmointikielenä. LabVIEW:n historia ulottuu 20 vuoden taakse, jolloin ohjelmankehitysympäristön kehittäminen on aloitettu. Uusin LabVIEW:stä julkaistu versio on 8.20.

LabVIEW:ssä ohjelmointi tapahtuu virtuaali-instrumenttien avulla. Ohjelmointiympäristö pitää sisällään valmiita virtuaali-istruumentteja, joita kutsutaan vi:ksi ja jotka suorittavat tietyn toiminnon. Avattaessa .vi huomataan, että se sisältää etupaneelin ja vuokaavion. Etupaneeliin (kuva 11) voidaan luoda oikeita mittalaitteita vastaavia kontrolli- ja indikaattoripaneeleja.



Kuva 11. Etupaneeli

Vuokaavioikkunan (kuva 12) puolella toteutetaan itse ohjelman suorittamat toiminnot. Ohjelman tekeminen suoritetaan paleteissa valmiina olevilla kuvakkeilla. Kuvakkeiden terminaalit yhdistetään langoitustyökalun avulla.



Kuva 12. Vuokaavioikkuna.

Ohjelmassa vi:lle vietään tietä vasemmalla puolella oleviin kontrolliterminaaleihin. Esimerkiksi Sine Waveform.vi (kuva 12) generoi sinisignaalin, jolle asetetaan taajuus- ja amplitudi-kontrollilla signaalin generoinnin tarvitsemat arvot. Vasteena vi:n oikealta puolelta saadaan generoitu signaali.

Ohjelmointiympäristö sisältää laiteajureita ja lisää niitä on saatavissa ohjelman tekijän, National Instrumentsin sivuilta. National Instruments valmistaa laajan valikoiman mittalaitteita ja näin ollen ajureitakin on helposti saatavilla kaikenlaisiin malleihin.

## 7 OHJELMAN TOTEUTTAMINEN

Ohjelman suunnitteleminen aloitettiin miettimällä, millainen suodatin työssä aiotaan toteuttaa. LabVIEW sisältää omat suodatinfunktiot, mutta työssä päädyttiin tekemään oma suodatin. Suodattimen tyyppiä valittiin kaistanpäästösuodin, jossa käytetään rectangle-ikkunointia.

### 7.1 Signaalin A/D-muunnos

PC-tietokoneen äänikortin A/D-muunnin muuntaa analogisen signaalin digitaaliseen muotoon. Vasteena muodostuu yksiulotteinen taulukko, joka on samanpituinen kuin näytteenotomääräksi on valittu Sound Input Configure-funktiossa (liite 2). Funktiossa alustetaan näytteistystaajuus, kanavien lukumäärä, sekä bittimäärä yhtä näytettä kohden. Näytteistystaajuutena käytettiin 22050 näytettä sekuntia kohden eli noin 22 kHz. Signaalin syötetään PC:n äänikortille monosignaalin, jolloin kanavien määräksi valittiin yksi (1). Lisäksi funktiossa asetettiin bittimääräksi 16 b/s näytettä kohden.

Näytteiden lukumääräksi asetettiin 1500 näytettä kanavaa kohden, jolloin ohjelma pystyi näytteistämään signaalin jatkuva-aikaisesti virheettömästi. Näytteiden lukumäärää kasvatettaessa laitteistolta vaaditaan lisää muistia. Jatkuva-aikainen näytteistäminen käyttää kiertävää puskuria, jolloin puskuria päivitetään koko ajan. Funktiossa voidaan myös valita päättyvä näytteistäminen, jolloin puskurin arvoa käytetään vain kerran.

Sound Input Configure-funktiossa näytteistetään data aikaisemmassa funktiossa määritellyillä spesifikaatiolla ja lähetetään puskurimuistiin odottamaan datan lukemista.

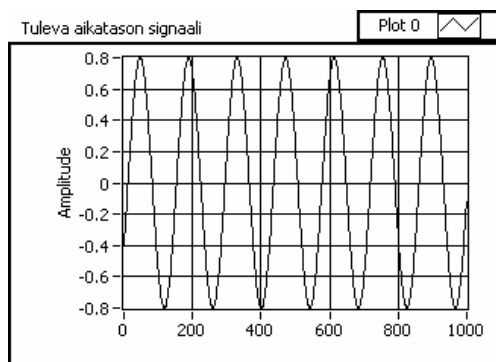
Puskurista tieto siirretään Sound Input Read-funktiolle. Funktio lukee puskurimuistiin kirjoitetun datan. Puskurista luettu data sisältää informaation näytteistetyistä signaaleista, näytteiden välisen ajan sekunteina, joka on näytteistystaajuuden käänteisarvo, ja aikaleiman ensimmäisestä luetusta näytteestä.

Yksiulotteinen taulukko erotetaan tietueesta Index Array-funktion avulla. Erotettu yksiulotteinen taulukko sisältää informaation aikatason signaalista. Taulukon solujen määrä viittaa signaalista otettavaan näytteiden määrään. Signaali saatiin siis sellaiseen muotoon, jonka jälkeen voitiin aloittaa signaalin suodattamisen vaatimat funktiot.

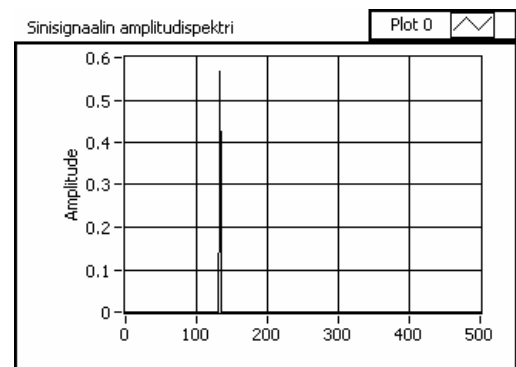
```
-0,50324 -0,76829 -0,75796 -0,47595 -0,02340 0,43757 0,74172
```

Kuva 13. Yksiulotteinen taulukko sinisignaalista.

Taulukkotiedosta saatiin muodostettua indikaattori, jossa taulukkotieto voitiin esittää graafisesti. Näyttö vastaa oskilloskoopin näyttöä, jossa signaali on esitetty aikatasossa (kuva 14a).



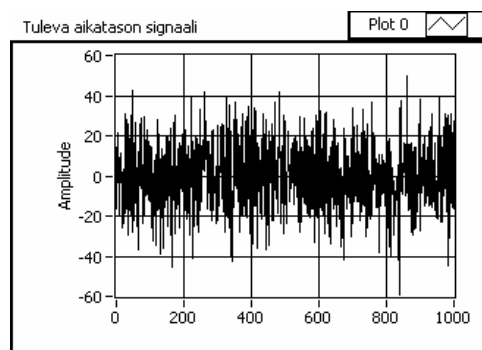
Kuva 14a. Taulukkotiedosta muodostettu kuvaaja.



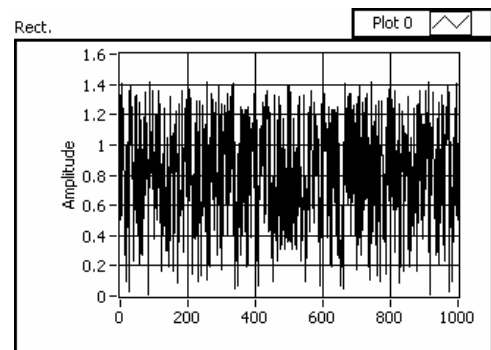
Kuva 14b. Sinisignaalin amplitudispektri.

Tulevalle aikatason signaalille tehtiin Fourier-muunnos, jolloin siitä nähtiin kaikki signaalissa mukana olevat taajuudet. Vasteksi sinisignaalin Fourier-muunnoksesta saatiin kuvaaja (kuva 14b). Amplitudispektrifunktio poistaa Fourier-muunnoksesta syntyvän identtisen osan, jolloin taulukon pituus puolittui.

Suodattimen päästökaistaa tutkittaessa ohjelmassa käytettiin kohinan Fourier-muunnosta. Kohinan sisältää kaikkia taajuuksia (kuva 15a ja 15b) ja sen avulla on selvästi nähtävissä suodattimen ominaisuudet.



Kuva 15a. Kohina aikatasossa.

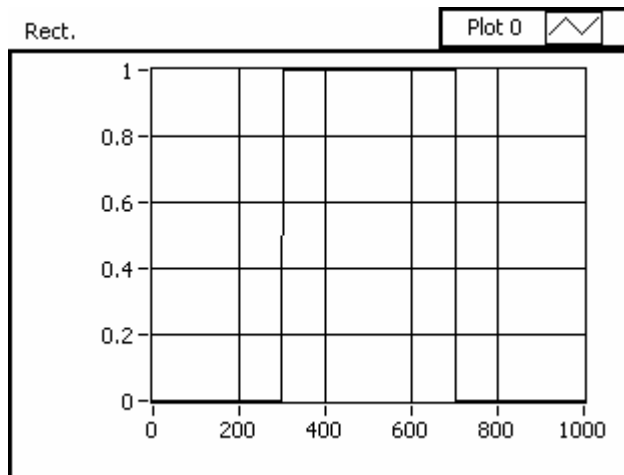


Kuva 15b. Kohina taajuustasossa.

## 7.2 Suodattimen toteuttaminen

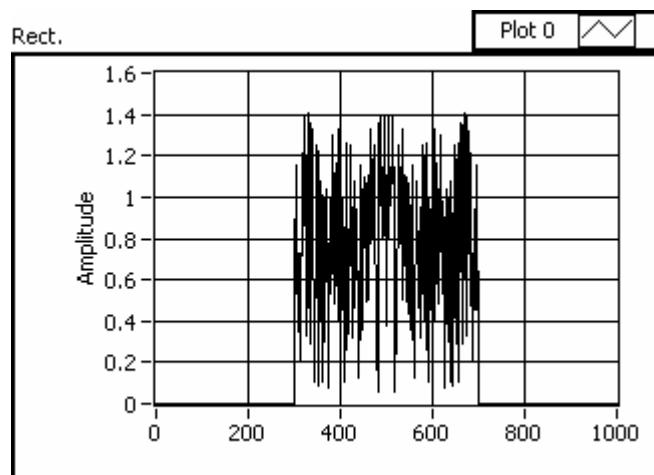
Rectangle-ikkuna toteutettiin for-silmukan avulla. Silmukalle alustettiin oma yksiulotteinen taulukko, jossa solut saivat arvokseen yksi (1). Silmukkaan toteutettiin ehto, jonka ollessa tosi taulukko saa arvokseen yksi. Vastaavasti ehdon ollessa epätosi taulukon arvo kerrottiin nolalla (0). Ehto tutki silmukan pyörähdysten lukumäärää ja kertoi annettujen arvojen perustella taulukon arvon luvulla nolla tai yksi. Vertailtava arvo ehdolle annettiin liukusäädinkontrollin avulla. Koska kyseessä oli kaistanpäästösuodatin, taulukon alkupään ja loppupään arvo kerrottiin arvolla nolla. Näin ollen saatiin muodostettua rectangle-ikkuna (kuva 16).





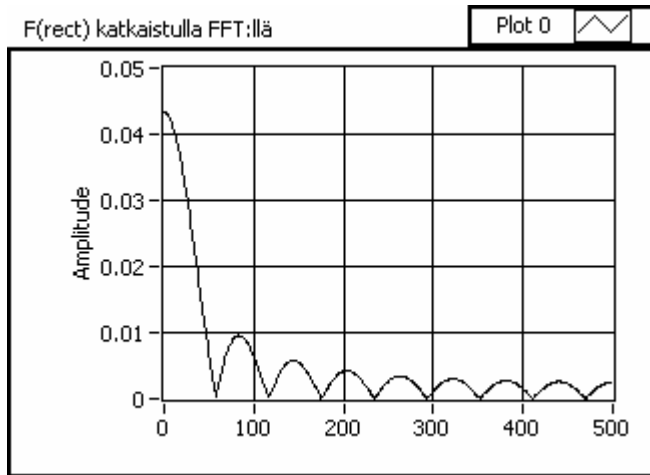
*Kuva 16. Rectangle-ikkuna.*

Kerrottaessa kohinasignaalin spektri muodostetulla rectangle-ikkunalla saadaan päästökaista (kuva 17). Kuvasta on selvästi havaittavissa läpipääsevä taajuuskaista.



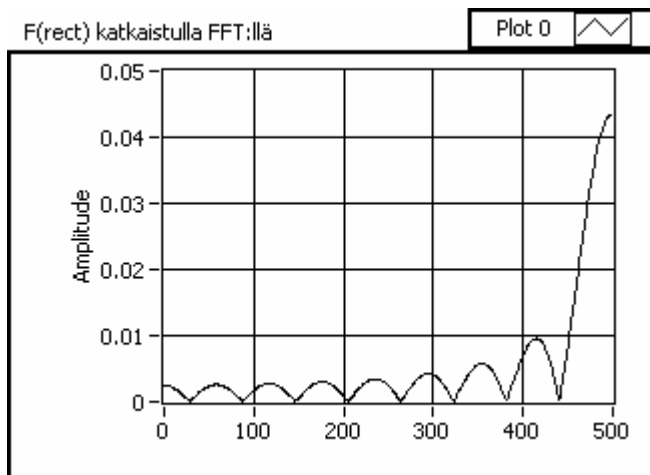
*Kuva 17. Kohinasignaalin amplitudispektri kerrottuna rectagle-ikkunalla.*

Taajuustasossakaan ei kuitenkaan voida suoraan asettaa signaalista muodostuvan taulukon arvoja nolaksi. Kyseisessä toimenpiteessä taajuudet vääristyvät. Vääristyminen estetään suorittamalla rectangle-funktiolle Fourier-muunnos. Vasteeksi toimenpiteestä pelkälle rectangle-ikkunalle tehdystä Fourier-muunnoksesta saatiin sinc-funktion itseisarvon puolikaan mukainen vaste (kuva 18).



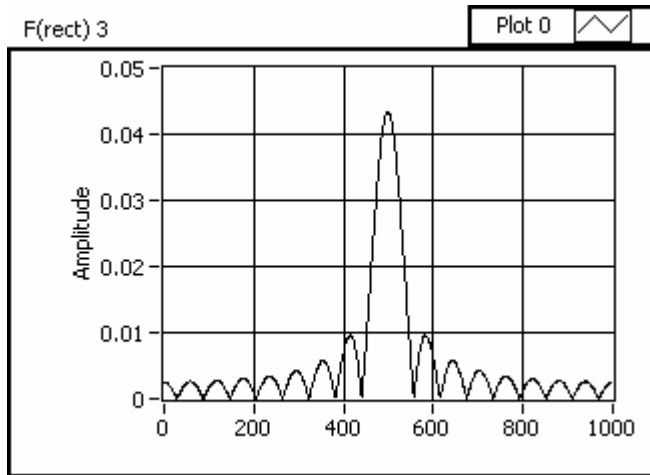
Kuva 18. Rectangle-ikkunan FFT-muunnos.

LabVIEW:n amplitudispektrifunktio katkaisee peilautuvan puolen pois FFT-muunnoksesta, jonka vuoksi rectangle-funktiosta saatava vaste ei tässä tapauksessa ole oikeanlainen. Katkaistun osan ollessa peilautuva taulukosta muodostetaan toinen taulukko, joka käännetään vastakkaiseksi. Voidaan ajatella, että peilautuva osa ulottuisi x-akselilla origon negatiiviselle puolelle (kuva 19).



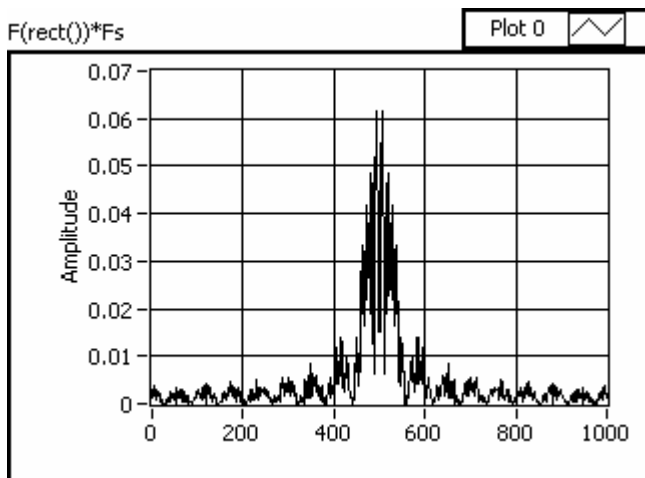
Kuva 19. Käännetty taulukko.

Tämän toimenpiteen jälkeen taulukot yhdistettiin ja vasteeksi saatiin sinc-funktion itseisarvo (kuva 20). Sinc-funktio on vakio, ja se muuttui ainoastaan muutettaessa rectangle-ikkunoinnin kokoa. Täten ohjelman suoritusta voitaisiin keventää suorittamalla rectangle-ikkunoinnin Fourier-muunnoksen laskeminen ainoastaan kerran ohjelman suorituksen alussa. Arvo kuitenkin on laskettava uudelleen, jos rectangle-ikkunan kokoa muutetaan.



Kuva 20. Sinc-funktion itseisarvo.

Saadulla sinc-funktiolla kerrottiin PC:n äänikortin A/D-muuntimen kautta siirretty aikatasoinen signaali, joka oli muunnettu taajuustasoon. Kuvassa 21 nähdään vaste kohinasignaalin amplitudispektristä kerrottuna rectangle-ikkunan Fourier-muunnoksella.



Kuva 21. Kohinan amplitudispektri kerrottuna rectangle-ikkunan Fourier-muunnoksella.

Itse signaalin suodattamiseen tarvittavat funktiot oli nyt suoritettu. Signaalin siirtäminen takaisin A/D-muuntimen kautta analogiseksi signaaliksi vaati IFFT-muunnoksen. Tästä muodostui ongelma tämän työn suorittamisessa, josta on kerrottu työn tulokset -osiossa. IFFT on käänteinen muunnos FFT-muunnokselle. LabVIEW:ssä olevan valmiin IFFT-funktion avulla taajuustasoinen signaali saadaan muunnettua aikatasoon, jonka jälkeen siitä saatu vaste voidaan kirjoittaa äänikortin D/A-muuntimelle.

## 8 TESTAAMINEN

Ohjelman testaamisessa tarvittiin apuvälineinä signaaligeneraattoria ja oskilloskooppia. Signaaligeneraattoriksi valittiin laite, jolla oli mahdollista syöttää kohinasignaalia. Signaali syötettiin PC-tietokoneen äänikortin MIC-IN -liitännän kautta 10 k $\Omega$ :n potentiometrin läpi. Ilman potentiometriä signaalin amplitudi oli liian suuri.

Ulostulevan signaalin mitattiin digitaalioskilloskoopin avulla. Oskilloskoopin mittapäät kytkettiin PC-tietokoneen äänikortin stereolähtöön. Signaaligeneraattorin avulla syötettiin sinisignaalia ja satunnaista kohinasignaalia. Sinisignaalia syötettäessä huomattiin, että signaali alkoi vaimeta noin 8 kHz:ssä, joten korkeampien taajuuksien käsitteleminen ei PC:n äänikortin avulla ole mahdollista.

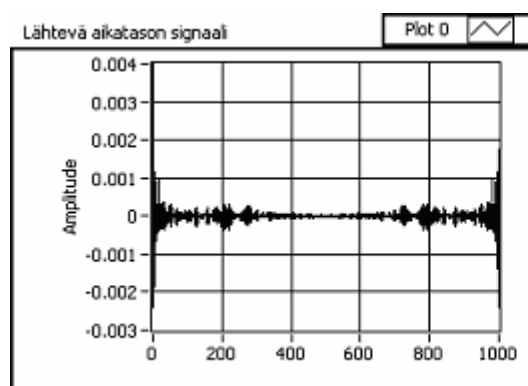
Kohinasignaalin avulla testattiin suodattimen päästökaistaa, jolloin ohjelman avulla taajuustasossa on selvästi nähtävissä kaistapäästösuodattimen päästökaista ja estokaistat. Ulostulevan signaalin oikeellisuutta testattiin sinisignaalin avulla.

## 9 TYÖN TULOKSET

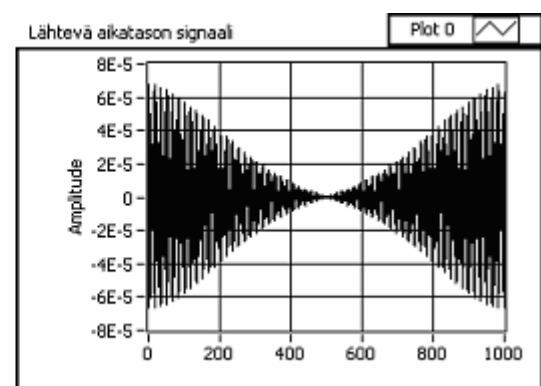
Työssä suodattimen ominaisuuksia mitattiin syöttämällä aikatason signaalia PC:n äänikortin A/D-muuntimen kautta ohjelmointiympäristöön. Signaalin syöttäminen tapahtui potentiometrin kautta, koska signaaligeneraattorin amplitudia ei saatu säädettyä tarpeeksi pieneksi PC:n äänikortin A/D-muuntimen ollessa suunniteltu mikrofonin tuottamaa pientä amplitudia varten.

Tehtävänannossa määritelty tutkimuksen aihe, voidaanko LabVIEWissä olevilla valmiilla VI-funktiolla toteuttaa digitaalinen signaalin suodatin käyttäen PC:n äänikorttia sai vastauksen. Ei voida. Ongelmaksi muodostui LabVIEW:ssä valmiina oleva IFFT-funktio. Funktio ei ymmärrä sille amplitudispektrifunktion syöttämää peilautuvan osan katkaisevaa taulukkoa. IFFT-funktio vaatii muunnoksesta myös identtisen puolen mukaan, sekä myös tiedon signaalin vaiheesta, jota amplitudispektri ei kerro.

PC-tietokoneen äänikortin ulostulosta mitattu aikatason kohinasignaali (kuva 22a) ei ollut sellainen kuin se pitäisi olla. Leveällä päästökaistalla ulostulevan signaalin pitäisi olla vastaava kuin syötteenä käytetty signaali. Vasteen oikeellisuus oli helppo todeta käyttämällä sinisignaalia syötteenä (kuva 22b).



Kuva 22a Lähtävä aikatason kohinasignaali



Kuva 22b Lähtävä aikatason sinisignaali.

## 10 YHTEENVETO

Työssä oli tarkoitus selvittää, voiko LabVIEW:ä käyttää tietoliikennetekniikan signaalinkäsittelysovelluksiin valmiiden funktioidensa avulla. Työn tavoitteena oli toteuttaa kaistanpäästösuodatin, jonka kaistanleveyttä ja keskitaajuutta oli mahdollista muuttaa. LabVIEW on tarkoitettu erityisesti elektroniikan mittaussovelluksien toteuttamiseen. Sen avulla ohjelmoiminen on nopeaa valmiiden funktioidensa ansiosta.

Työssä saatiin toteutettua ohjelma, jossa signaali siirretään PC-tietokoneen äänikortin kautta ohjelmointiympäristöön. Signaalille suoritettiin FFT-muunnos, jolloin signaali saatiin aikatasoisesta signaalista taajuustasoiseksi signaaliksi. Taajuustasossa on erotettavissa kaikki signaalin sisältämät taajuudet. Aikatasossa signaalin sisältämiä taajuuskomponentteja ei voida tarkasti havaita. Taajuustasossa signaalille suoritettiin kertominen suodatinfunktiolla. Suodatinfunktiona käytettiin rectangle-ikkunaa, jolle suoritettiin myös FFT-muunnos. Rectangle-ikkunalle suoritettun FFT-muunnoksen impulssivasteeksi saatiin sinc-funktion itseisarvo. Molemmat muunnetut signaalit voidaan edellä mainittujen toimenpiteiden jälkeen kertoa keskenään. Aikataason konvoluutio on taajuustason kertolasku. Tämän jälkeen vasteelle suoritettiin käänteinen FFT-muunnos (IFFT), jolloin signaali saatiin palautettua aikatasoiseksi.

Työssä perehdyttiin näin ollen signaalin käsittelyyn, joka piti sisällään tässä tapauksessa signaalin muuttamisia aika- ja taajuustason välillä ja myös suodattimen toteuttamistapoihin perehtymistä. Työ oli haasteellinen signaalin käsittelyn osalta.

## LÄHTEET

- 1 Digital Signal Processing. [WWW-dokumentti]  
<<http://www.dsptutor.freeuk.com/intro.htm>>. (Luettu 13.10.2006.)
- 2 Määttä E. Signaalien luokittelu. Luentomoniste.
- 3 The Athena Group, Inc, Jon Mellot. Hands-On Digital Signal Processing. 1998  
ISBN 0-07-912965-X
- 4 John G. Proakis Dimitri G. Manolakis. Digital Signal Processing. New Jersey: 1996  
ISBN 0-13-394338-9
- 5 C.S. Burrus and T.W Parks. DFT/FFT and Convolution Algorithms. Canada: 1985  
ISBN 0-471-81932-8
- 6 The Scientist and Engineer's Guide to Digital Signal Processing  
By Steven W. Smith, Ph.D. [WWW-dokumentti]  
<<http://www.dspguide.com/ch6.htm>>. (Luettu 15.11.2006.)

## LIITTEIDEN LUETTELO

LIITE A	MATLAB-ohjelma
LIITE B	LabVIEW-ohjelman vuokaavio



```
t = 0:0.001:25;  
x = sin(2*pi*50*t) + sin(2*pi*120*t);
```

```
y = x + 2*randn(size(t));  
plot(y(1:50))  
title('Noisy time domain signal')
```

```
Y = fft(y,256);
```

```
Pyy = Y.*conj(Y)/256;  
f = 1000/256*(0:127);  
plot(f,Pyy(1:128))  
title('Power spectral density')  
xlabel('Frequency (Hz)')
```

