

Jukka-Pekka Kujala

Äänen analysointi, esittäminen graafisesti ja vertailu

**Opinnäytetyö
CENTRIA AMMATTIKORKEAKOULU
Mediatekniikan koulutusohjelma
Huhtikuu 2014**

TIIVISTELMÄ OPINNÄYTETYÖSTÄ

Yksikkö Ylivieska	Aika Huhtikuu 2014	Tekijä/tekijät Jukka-Pekka Kujala
Koulutusohjelma Mediatekniikka		
Työn nimi Äänen analysointi, esittäminen graafisesti ja vertailu		
Työn ohjaaja Hannu Puomio	Sivumäärä 26	
Työelämäohjaaja Centria sovelletun elektroniikan laboratorio / Ritva Saviluoto		
<p>Työn pääasiallinen tarkoitus oli tutkia millä keinoilla ääntä voidaan ohjelmallisesti muokata ja vertailla liittyen toimeksiannetun Sheemu-ohjelmiston kehittämiseen. Työ koostui mahdollisesti hyödyllisten ohjelmien etsimisestä ja toiminnan testauksesta sekä teoritiedon keräämisestä äänen digitaaliseen ja analogiseen käsittelyyn liittyen. Teoritiedot käsittivät äänen rakenteen perusteita, sekä äänen graafiseen esittämiseen liittyvien algoritmien tutkimista.</p>		
Asiasanat Analysointi, FFT, MIDI, ääni.		

ABSTRACT

CENTRIA UNIVERSITY OF APPLIED SCIENCES Ylivieska	Date April 2014	Author Jukka-Pekka Kujala
Degree programme Mediatechnology		
Name of thesis Sound analyzing, comparison and graphic displaying		
Instructor Hannu Puomio		Pages 26
Supervisor Centria embedded systems laboratory / Ritva Saviluoto		
<p>The main focus of thesis was to study how we can analyse and compare different sounds with computer software. Thesis consisted of looking useful programs and methods that could be used in developing the given application. This also required studying of sound theory in general and taking a closer look in some of the algorithms used in displaying the analysed sound.</p>		

<p>Key words Analyzing, FFT, MIDI, sound.</p>
--

KÄSITTEIDEN MÄÄRITTELY

A/D	Analog to Digital
Amplitudi	Signaalin voimakkuus
D/A	Digital to Analog
DFT	Discrete Fourier Transform
FFT	Fast Fourier Transform
MIDI	Musical Instrument Digital Interface

**TIIVISTELMÄ
ABSTRACT
KÄSITTEIDEN MÄÄRITTELY
SISÄLLYS**

1 JOHDANTO	1
2 TIETOPERUSTA, MENETELMÄT JA AINEISTO	2
2.1 Audiosignaalin käsittely	2
2.2 Digitaalisen äänen viestiketju	3
2.3 Suodatus ja laskostuminen	4
2.4 Kvantisointi	5
2.4.1 Kvantisointivirhe	7
2.4.2 D/A-muunnos	8
2.5 Fourier analyysi ja synteesi	9
2.6 Perustaajuuden etsintä	11
3 MIDI TIEDOSTON LUKEMINEN JA TOISTAMINEN	13
3.1 Historiaa	13
3.2 Viestit ja kanavat	14
3.3 General MIDI	17
4 LÖYDETYT OHJELMISTOT	20
4.1 FFT-guitar tuner	20
4.2 Simple Audio Out Oscilloscope and Spectrum Analyzer	21
4.3 FFT of waveIn audio signals	22
4.4 C# Midi toolkit	23
4.5 Frequency analyzer	24
5 TULOKSET JA RATKAISUEHDOTUKSET	25
6 LÄHTEET	26

1 JOHDANTO

Keväällä 2013 sain toimeksiannon Sheemu-projektiin liittyen. Sheemu on tablet-laitteelle kehityksessä oleva ohjelmisto jonka avulla voitaisiin harjoitella haluttua musiikkikappaletta. Tablet-laite näyttäisi valitun musiikkikappaleen nuotit, sekä vertailisi mikrofoniin lähtevän kautta tulevaa signaalia nuotiston arvoihin. Musiikkikappaleen mukaan voitaisiin valita haluttu instrumentti tai vaikkapa laulu harjoiteltavaksi. Ohjelmisto toistaisi itse jäljelle jäävät soittimet, laulun tms. sekä näyttäisi miten hyvin harjoittelija pysyy kappaleen vaatimassa sävelessä jonkilaaisella graafisella elementillä. Tällä hetkellä projekti taitaa olla hieman jäissä, koska työn tilaajaan ei ole saatu yhteyttä monista yrityksistä huolimatta.

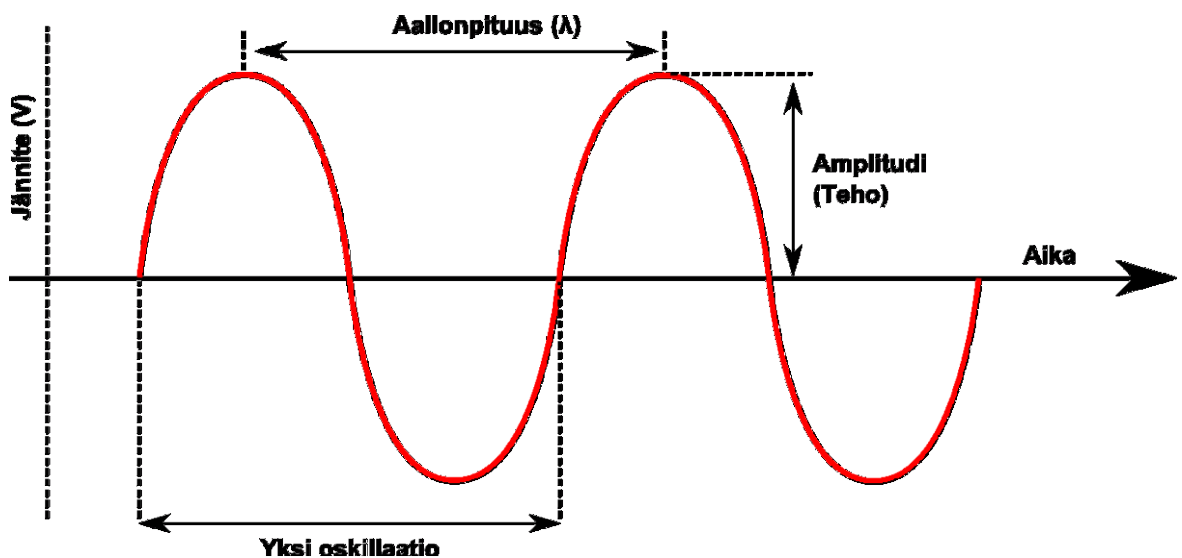
Ensimmäiseksi toimeksiantoon liittyen rupesin etsimään jo valmiiksi kehiteltyjä ohjelmistoja tai koodinpätkiä, joita olisi mahdollisimman helppo hyödyntää. Tämä aiheutti vaatimuksen ainakin ohjelmiston lähdekoodin vapaaseen saatavuuteen. Käyttökelpoisten ohjelmien löytäminen osoittautuikin yllättävän hankalaksi. Useimmiten löytynyt ohjelma oli ohjelmoitu vuosia sitten, jolloin se ei enää toiminut halutulla tavalla. Ohjelmistoista ei saanut lähdekoodia, tai se oli saatavilla vain osittain. Onneksi myös käyttökelpoisiakin ohjelmia löytyi.

Toinen vaihe loppu työssäni käsitteli teorian tiedon keräämistä ja kokoamista järkeväksi kokonaisuudeksi. Teorian tiedot käsittelevät ääneen muokkaukseen ja graafisen esityksen keinoja ja menetelmiä. Äänen käsitystä fyysisenä ilmiönä sekä sen matemaattista ja ohjelmallista käsittelyä mm. Fourier-muunnosten avulla. Työssä käydään myös läpi MIDI-tiedostojen historiaa, sekä sen mahdollista hyödyntämistä kyseisessä projektissa.

2 TIETOPERUSTA, MENETELMÄT JA AINEISTO

2.1 Audiosignaalin käsittely

Mitä ääni on? Meillä jokaisella on käytännön kokemuksia erilaisista äänistä jokapäiväisessä elämässä, mutta mitä se fysiikan kannalta oikein on? Miten ääntä on mahdollista tutkia ja havainnollistaa erilaisia laitteita hyväksikäyttäen? Yksinkertaistaen ääni on aaltoliikettä josta ihmisen korva havaitsee äänen paineessa tapahtuvat muutokset. Tarkastellaan tätä ilmiötä hieman tarkemmin oheisen kuvion 1 avulla. (Peltonen, Perkkiö & Vierinen. 2012, 126-129.)



KUVIO 1. Ääni aaltoliikkeenä (Mukaiillen Peltonen, Perkkiö, Vierinen 2012, 87,126)

Oheisessa kuvassa tarkastellaan aaltoliikettä jännitteen ja ajan funktiona. Koska ääni on aaltoliikettä, niin siitä voidaan määritellä: taajuus, aallonpituus sekä nopeus. Jos tiedetään kuinka paljon on taajuuden yhden kierron eli ns. oskillaation värähdysaika, voidaan siitä laskea ääniaallon taajuus kaavalla: $F = \frac{1}{T}$ Esim. jos $T = 0,5 \text{ m}$ saadaan taajuudeksi n. 2000Hz. Jos äänen nopeus ja taajuus tiedetään, voidaan myös aallonpituus ratkaista aaltoliikkeen perusyhtälöllä: $V = \lambda F$. (Peltonen, Perkkiö & Vierinen. 2012, 126-129.)

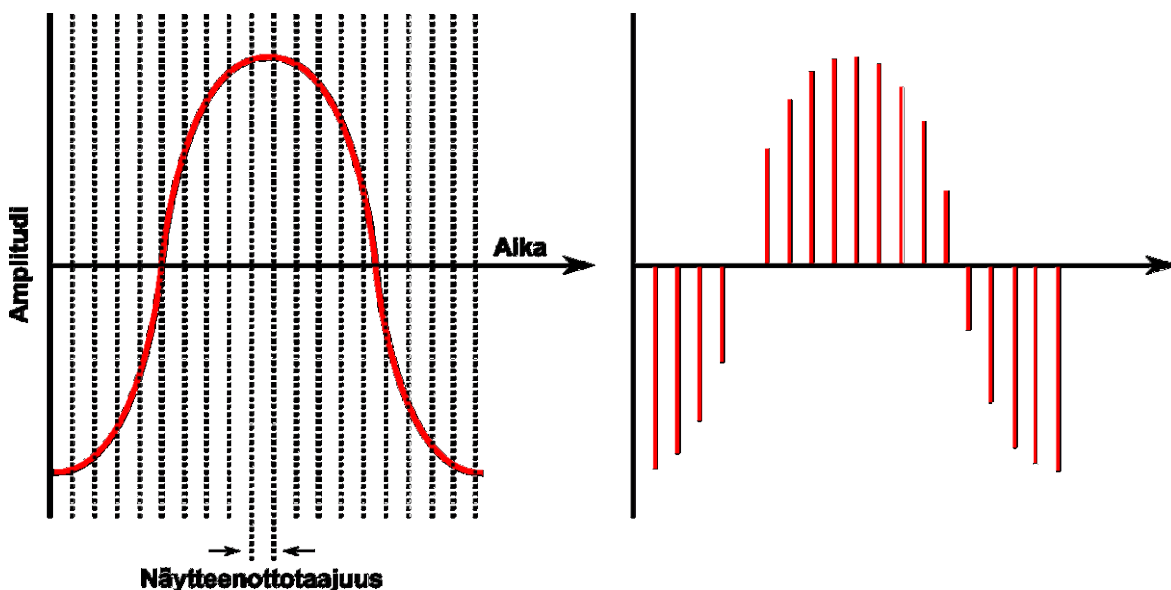
2.2 Digitaalisen äänen viestiketju

Kuviossa 2 nähdään tyypillinen digitaalisen äänityksen viestiketju. Ensimmäiseksi analoginen signaali menee A/D muuntimen läpi. A/D-muuntimessa jatkuvasti vaihteleva analoginen signaali muunnetaan sarjaksi näytteitä. Nämä näytteet ovat osia analogisesta signaalista josta on mitattu analogisen signaalin arvoja jopa tuhansia kertoja. Jokaista näytettä vastaa numero. Tämä näytteiden sarja koodataan sellaiseen muotoon, että se on mahdollista tallentaa tai lähettää eteenpäin (kanavakoodaus). Äänitettä toistettaessa koodaus puretaan ja lähetetään virheenkorjauksen läpi. Virheenkorjauksen avulla korjataan häviötä, joka on muodostunut A/D muunnoksessa. Jokainen virhe ajoituksessa tai näytteiden arvoissa yritetään korjata ennen lähettämistä eteenpäin D/A-muuntimelle. D/A-muunnin muuttaa digitaalisen signaalin takaisin jatkuvaksi analogiseksi signaaliksi, jolloin se voidaan toistaa. (Rumsey & McCormick 1999, 185.)



Tyypillinen esimerkki digitaalisen äänityksen viestiketjusta

KUVIO 2. Digitaalisen äänen viestiketju (Rumsey & McCormick 1999, 185)

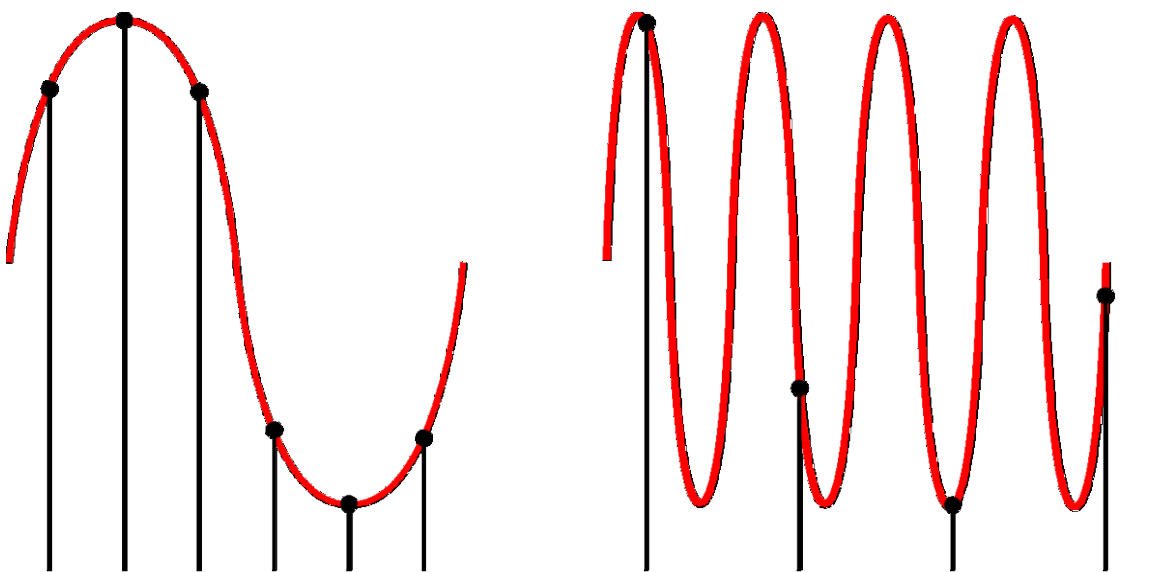


KUVIO 3. Digitaalisen äänen näytteistys (Rumsey & McCormick 1999, 185)

Analoginen signaali näytteistetään käyttäen tiettyä näytteenottotaajuutta (KUVIO 3). Tämän avulla saadaan sarja värähdyksiä, joiden amplitudit osoittavat analogisen signaalin arvot tietyinä ajanjaksona. Näytteistykseen näytteet voidaan ajatella amplitudi pulsseina joiden avulla myöhemmin muodostetaan digitaalinen signaali takaisin analoogiseksi signaaliksi. Jotta analooginen signaali saataisiin talteen mahdollisimman samana on näytteitä otettava useita kertoja sekunnissa. Shannonin matemaattisen teorian mukaan jokaista näytettä kohden on otettava kaksi näytettä. Tällöin näytteenottotaajuuden on oltava vähintään kaksinkertainen verrattuna tallennettavan äänisignaalin maksimitaajuuteen. (Rumsey & McCormick 1999, 186.)

2.3 Suodatus ja laskostuminen

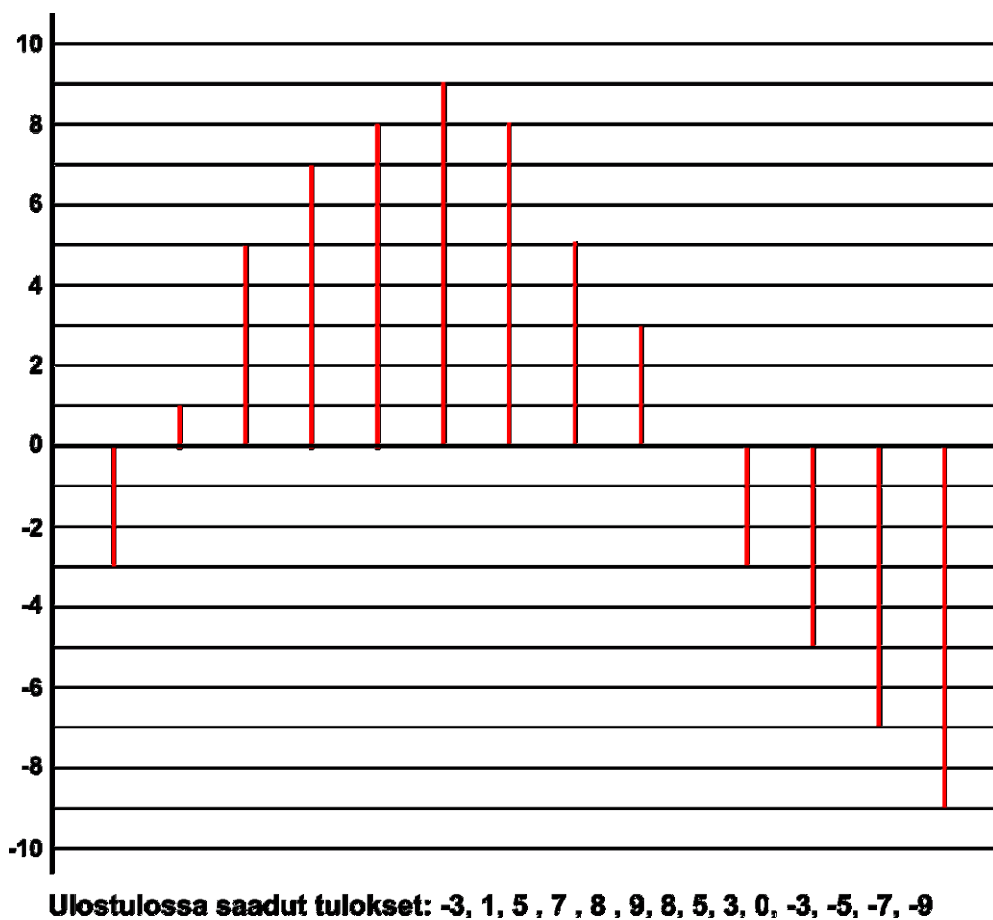
Näytteenottotaajuuden on siis oltava riittävän suuri jotta virheitä välttyttäisiin. Oheisen kuvan avulla voidaan yksinkertaisesti osoittaa miten liian pieni näytteenottotaajuus aiheuttaa signaalin laskostumista. Signaalista katoaa tärkeää informaatiota jos näytteenottotaajuus on liian pieni. (Rumsey & McCormick 1999, 185-188)



KUVIO 4. Laskostuminen (Rumsey & McCormick 1999, 185)

2.4 Kvantisointi

Kun audiosignaali on näytteistetty, se on muunnettava sarjaksi numeroita. Tätä muunnosta kutsutaan kvantisoinniksi (KUVIO 5). Näytteistämisen avulla saadaan sarja pulsseja vaihtelevilla amplitudiarvoilla. Amplitudiarvot muunnetaan kvantisoinnin avulla numeerisiksi arvoiksi verraten miten kaukana amplitudit ovat asteikon nollatasosta. Jokainen näyte saa arvon etukäteen valitun asteikon mukaan. Esimerkissä asteikkona käytetään arvoja 1-10 positiiviseen ja negatiiviseen suuntaan. Jokaiselle näytteelle määritetään tasaluku, eikä desimaaleja sallita. Kvantisoinnissa siis pyöristetään amplitudiarvot lähimpään arvoonsa sen mukaan mitä kvantisointiasteikkoa on käytetty. (Rumsey & McCormick 1999, 189-192.)

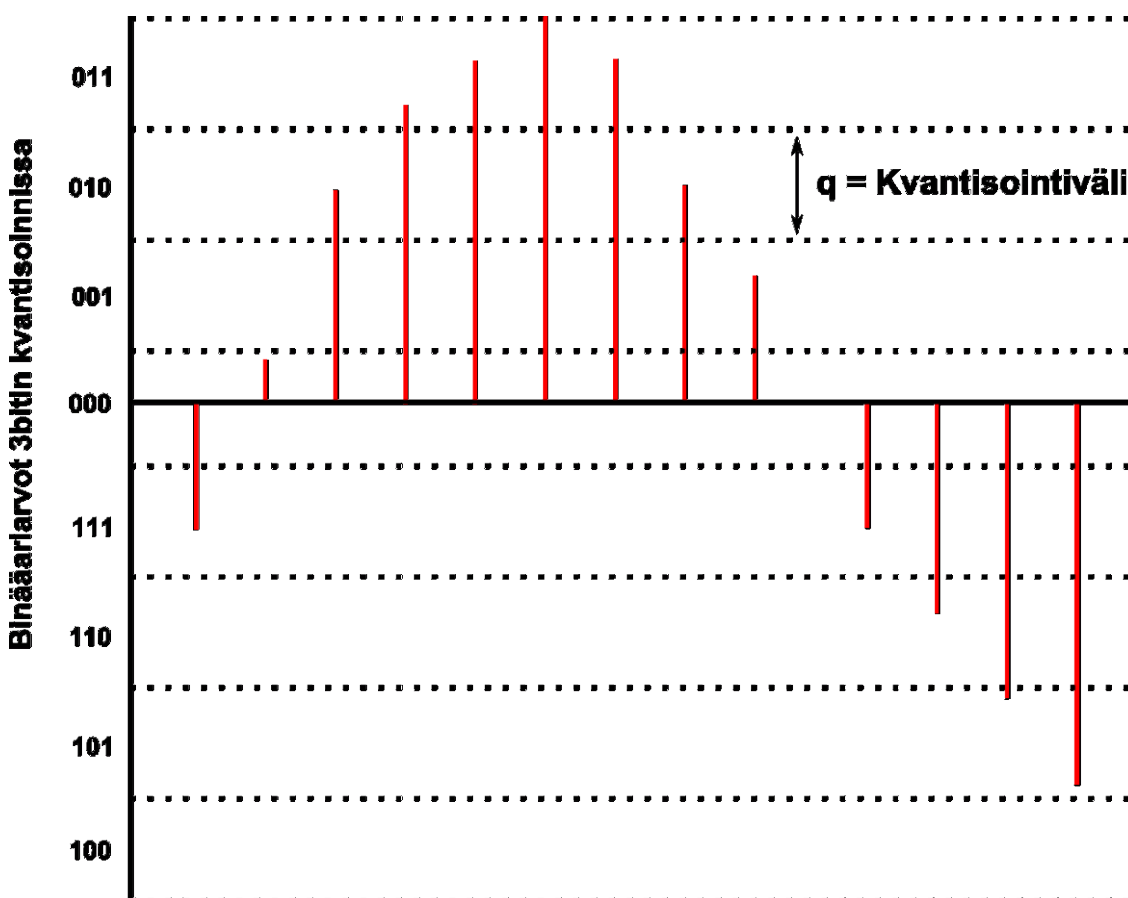


KUVIO 5. Digitaalisen äänen kvantisointi (Rumsey & McCormick 1999, 191)

Digitaalisessa äänenkäsittelyssä käytössä oleva numerojärjestelmä on binäämuotoinen desimaalilukujen sijaan. Binäärinkäytöstä saadaan monia hyötyjä, koska se mahdollistaa numeroiden esittämisen pelkästään kahden tilan muodoissa (on/off, tosi/epätosi, jännitettä,

jännitteetön) sekä mahdollistaa äänisignaalin tallentamisen samalla tavalla kuin mikä tahansa muu tietokoneen käsittelemä data. Desimaalilukujen kantaluku on 10, mutta binääriluvuissa se on 2. Yhtä binäärilukua kutsutaan bitiksi. Kahdella bitillä voidaan ilmaista neljä tilaa. (00,01,11,10). Mitä enemmän bittejä on käytössä sitä suurempi määrä tiloja voidaan esittää. Esim. $2^8 = 256$, $2^{16} = 65536$. (Rumsey & McCormick 1999, 189-192.)

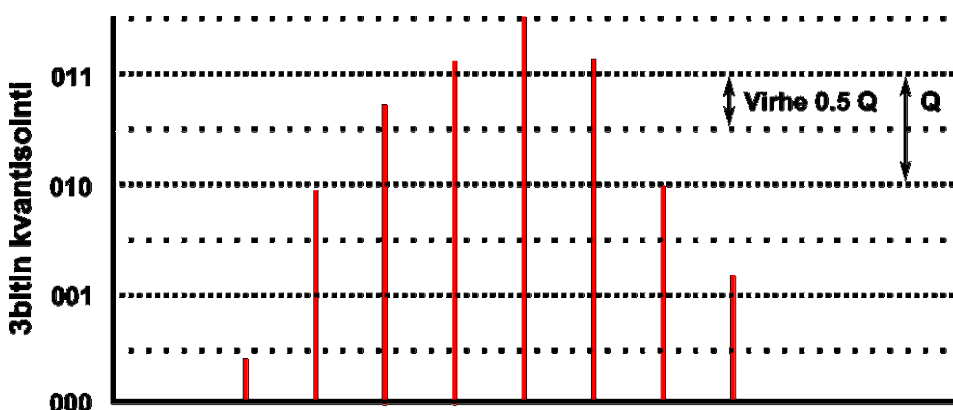
Kolmella bitillä voidaan osoittaa kahdeksan eri tilaa ja esimerkissämme on täten käytetty kolmen bitin kvantisointia (KUVIO 6).



KUVIO 6. Kvantisointiväli (Rumsey & McCormick 1999, 192)

2.4.1 Kvantisointivirhe

Koska harvemmin näytteistyksessä olevat näytteet ovat tarkkaan tietyillä amplitudeilla kvantiointiasteikon suhteen joudutaan ne pyöristämään lähimpään arvoon. Tätä virhettä kutsutaan kvantisointivirheeksi (KUVIO 7). Suurin mahdollinen kvantisointivirhe on puolet kvantisoinnin yhden asteikon välistä. Mikäli kvantisoinnin tarkastelun välejä lisätään, vähentää se myös virheiden määrää. (Rumsey & McCormick 1999, 192-195.)



KUVIO 7. Kvantisointivirhe (Rumsey & McCormick 1999, 193)

2.4.2 D/A-muunnos

Jotta digitaalinen äänitys voitaisiin toistaa se pitää konvertoida binäärimuotoisesta datasta takaisin analogiseksi signaaliksi. Perustason D/A-muunnos on esitelty kuviossa 8. Näytteistys muunnetaan jännitetasojen sarjaksi joka muistuttaa portaita. Tämä saavutetaan yksinkertaisilla muuntimilla jossa bittien tilat osoittavat jännitteiden amplitudit jokaisen lähdön mukaan. Tämä porrastuksen kaltainen näytejono uudelleennäytteistetään pienentämällä jokaisen pulssin leveyttä jotta se voitaisiin lähettää eteenpäin alipäästösuodattimelle. Alipäästösuodattimen suodatustaajuus on puolet näytteenottotaajuuden määrästä. Alipäästösuodattimen avulla näytteiden välit saadaan yhtenäiseksi, jolloin tuloksena syntyy sileä aaltomuoto. Uudelleennäytteistys on tarpeen, koska muuten suodattimen pyöristys aiheuttaisi korkeataajuuksisten ääninäytteiden amplitudi häviötä. (Rumsey & McCormick 1999, 198-199.)



KUVIO 8. D/A-muunnos (Rumsey & McCormick 1999, 198)

2.5 Fourier analyysi ja synteesi

Signaalien kuvaaminen ja käsittely on useasti tehokkaampaa, jos signaalit ja vasteet muunnetaan taajuusalueeseen, eli ne esitetään taajuuden funktiona. Useasti tämä tapahtuu Fourier-muunnoksen avulla. (Fourier Transform):

$$X(\omega) = F\{x(t)\} = \int_{-\infty}^{+\infty} x(t)e^{-j\omega t} dt \quad (1)$$

$$X(k) = F_d\{x(n)\} = \sum_{n=0}^{N-1} x(n)e^{-jk\left(\frac{2\pi}{N}\right)n} \quad (2)$$

joissa imaginääriyksikkö $j = \sqrt{-1}$. Vielä yleisemmin käytetty on Laplace muunnos, mutta signaalinkäsittelyssä Fourier muunnosta käytetään paljon laajemmin. Ensimmäinen kaava pätee aikajatkuviin tapauksiin ja toinen kaava aikadiskreetteihin. Jälkimäistä kaavaa kutsutaankin diskreetiksi Fourier-muunnokseksi (DFT, discrete Fourier Transform). Se on määritetty äärellisen mittaisella sekvenssillä (N) ja sen avulla laskutoimitukset toimivat erittäin tehokkaasti kun käytetään avuksi FFT-muunnosta (Fast Fourier Transform). Näiden kaavojen muunnoksia kutsutaan Fourier-analyysiksi. Kaavojen käänteiset muunnokset ovat:

$$x(t) = F^{-1}\{X(\omega)\} = \frac{1}{2\pi} \int_{-\infty}^{+\infty} X(\omega)e^{j\omega t} d\omega \quad (3)$$

$$x(n) = F_d^{-1}\{X(k)\} = \frac{1}{N} \sum_{k=0}^{N-1} X(k)e^{jk\left(\frac{2\pi}{N}\right)n} \quad (4)$$

Näiden kaavojen avulla taajuusalue-esitykset muuntuvat takaisin aikasignaaleiksi ja niiden muunnoksia kutsutaan Fourier-synteesiksi. Fourier muunnoksen huomattava etu on , että se muuntaa konvoluutio-operaation kertolaskuksi.

$$F\{x(t) * y(t)\} = X(\omega) * Y(\omega) \quad (5)$$

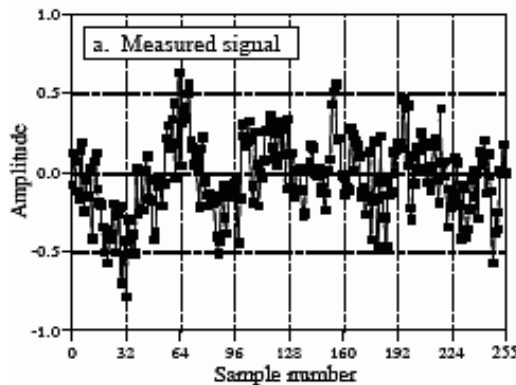
$$F_d\{x(n) * y(n)\} = X(k) * Y(k) \quad (6)$$

Tämä kertolasku on paljon yksinkertaisempi ja tehokkaampi laskea. Fourier-muunnosta ja sen käänteismuunnosta hyväksi käyttämällä voidaan konvuluution laskenta suorittaa kertolaskun avulla. (Karjalainen 2009, 34-35.)

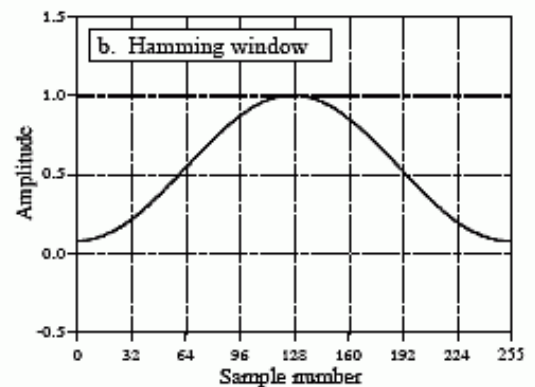
$$x(t) * y(t) = F^{-1}X(\omega) * Y(\omega) = F^{-1}\{F\{x(t)\} * F\{y(t)\}\} \quad (7)$$

$$x(n) * y(n) = F_d^{-1}\{X(k) * Y(k)\} = F_d^{-1}\{F_d^{-1}\{x(n)\} * F_d\{y(n)\}\} \quad (8)$$

2.6 Perustaajuuden etsintä

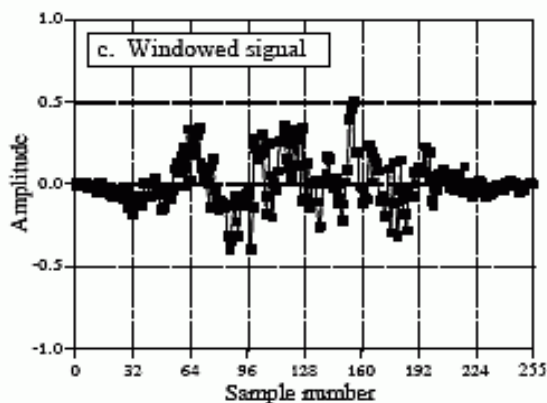


KUVIO 9. Näytesignaali (Smith 1997)

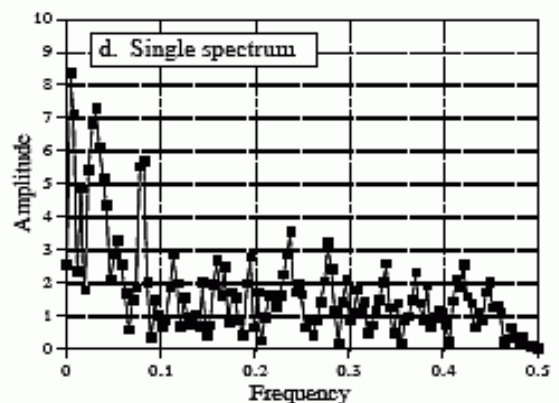


KUVIO 10. Hamming (Smith 1997)

Tutkittavan äänisignaalin sisältämien perustaajuuksien selvittäminen on yksi digitaalisen signaalinkäsittelyn perusteista. Kuvio 9 on 256 näytteen pituinen käyrä äänisignaalin amplitudi-arvoista. Kuvion 9 mukainen hajanainen amplitudiarvojen käyrä ei kuitenkaan ole vielä kovinkaan informatiivinen. Käsittelemme arvoja seuraavaksi kertomalla niitä Hamming window-menetelmän avulla (KUVIO 10). Tämän tuloksena saadaan 245 näytteen signaali jonka alku- ja loppupään amplitudearvot ovat tasaantuneet (KUVIO 11). (Smith 1997.)



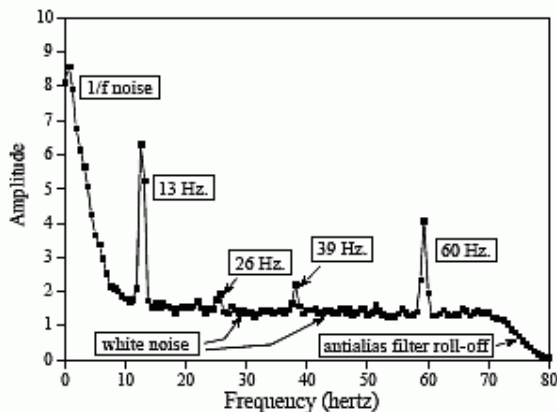
KUVIO 11. Ikkunoinnin tulos(Smith 1997)



KUVIO 12. DFT-muunnos (Smith 1997)

Jos tämän käyrän arvot muunnetaan DFT-algoritmin (Discreet Fourier Transform) avulla ei tulos ole kuitenkaan vielä hyödyllinen (KUVIO 12). Tämä johtuu siitä että 245 näytteen äänisignaali ei sisällä tarpeeksi informaatiota, jotta saataisiin käyttökelpoisia tuloksia.

Vaikka DFT:n käsittelemiä näytteitä lisättäisiin, se ei auttaisi siistimään lopputulosta. Näytteiden lisäys antaisi paremman resoluution taajuuden suhteen, mutta häiriöt olisivat siellä edelleen. Jotta saataisiin hyödyllisiä tuloksia on alkuperäisen signaalin näytteistystä lisättävä siten ettei taajuuden spektri kasva. Tällainen tulos saadaan pilkkomalla alkuperäinen signaali useisiin 256 näytteen lohkoihin. Jokainen lohko kerrotaan Hamming window-menetelmällä ja muunnetaan DFT-algoritmin avulla. Näin syntynyt taajuusspektri vielä keskiarvoistetaan jolloin saadaan 129 näytteen taajuuspekkti. Syntynyt lopputulos on huomattavasti käyttökelpoisempi. Häiriöt ovat suurimmilta osin poistuneet jolloin tärkein informaatio signaalin tutkimisen kannalta on helposti tarkasteltavissa. Tarkastellaan syntynyttä spektriä hieman tarkemmin. (KUVIO 13.) (Smith 1997.)



KUVIO 13. Lopullinen signaali (Smith 1997)

13hz kohdalla nähdään piikki amplitudissa ja pienemmät piikit 26hz ja 39hz kohdalla. 13hz kohdan piikistä käy ilmi signaalin perustaajuus, jolloin piikit 26hz ja 39hz kohdista ovat ns. ensimmäinen ja toinen harmoninen aalto. Lähemmin tarkasteltuna harmonisen aallon piikkejä pitäisi näkyä myös 52hz, 65hz, 78hz jne. kohdalla, mutta ne peittyvät valkoisen kohinan alle. (Smith 1997.) Tämän käsittelyn avulla saadaan siis selvitettyä mitatun signaalin perustaajuudet. Tätä menetelmää voidaan käyttää hyödyksi esim. kun halutaan vertailla soitettua säveltä ennalta määritetyn sävelen perustaajuuteen, verraten niiden samankaltaisuutta. Tähän menetelmään esim. kitaran viritysmittarit perustuvat. Ohjelmisto mittaa mikrofinin kautta tulevaa signaalia tietyn sävelen perustaajuuteen ja ilmoittaa ovatko ne samat, vai ylä- tai alavireessä.

3 MIDI TIEDOSTON LUKEMINEN JA TOISTAMINEN

3.1 Historiaa

Aluksi on syytä tarkastella miksi MIDI kehiteltiin. Aikaisemmin n.1950-luvulla studio saattoi koostua useistakin erilaisista laitteista. Tyypillisesti mukana oli esim. oskillaattori, pulssigeneraattori, kohinageneraattori, erilaisia suotimia, kaikulaitteita, amplitudi-, taajuus- ja rengasmodulaattoreita, nauhureita, oskilloskooppi, spektrianalysaattori, nauhanopeuden säätölaite ja mikseri. Laitteet oli kerätty useistakin eri lähteistä ja osa saattoi olla itse rakennettuja. Eri laitteiden liittäminen toisiinsa vaati elektroniikan tuntemusta ja useimmiten itse rakennettuja kytkentöjä.

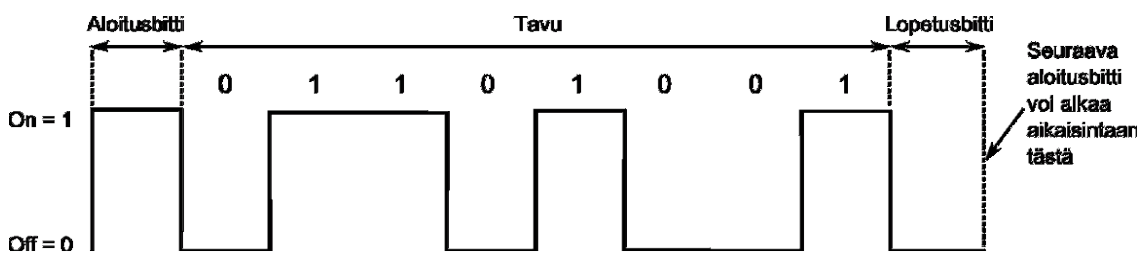
Vuonna 1964 valmistui ensimmäisenä pidetty syntesoiija josta käytettiin nimitystä Moog. Moog-syntesoiija sisälsi aiemmin erillisinä laitteina tarvittut komponentit kuten suotimet tai oskillaattorit. Kaikki tarvittut komponentit oli rakennettu moduleiksi jotka keskustelivat keskenään jänniteohjauksen avulla ja ne voitiin kytkeä toisiinsa yksinkertaisesti vain laitteen etupaneeliin kytketyillä kaapeleilla. Myöhemmin 1980-luvulla tehtiin uusi innovaatio, joka suuremmilta osin muistutti Moogin käyttöperiaatteita. Syntyi kokonaan ohjelmoitavissa oleva syntesoiija joka kuitenkin oli vain yksi osa suurempaa musiikkilaitteistoa. Sen avulla eri laitteiden mikroprosessorit saatiin keskustelemaan keskenään yhtenäisellä kieliopilla ja sanastolla. MIDI-kieli oli syntynyt. MIDI-kielen avulla saatiin luotua yhtenäinen musiikkijärjestelmä jonka avulla pystyttiin kontrolloimaan yhtenäisesti useampia eri laitteita. MIDI:stä muodostui myöhemmin yleinen musiikkilaitteiden tiedonsiirtostandardi, jonka jatkokehitystä tehdään edelleenkin.

(Hirvi & Tuominen 1995, 10-13.)

3.2 Viestit ja kanavat

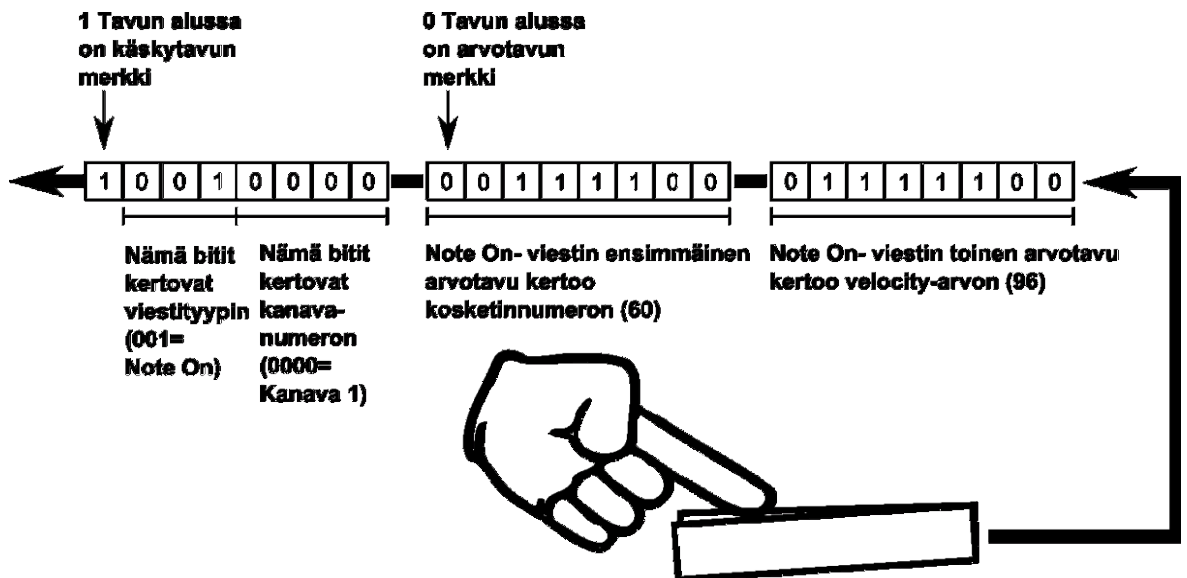
MIDI on siis musiikillista tiedonsiirtoa, tarkemmin ajateltuna sarjallista ja asynkronista tiedonsiirtoa. Sarjallisella tarkoitetaan sitä kuinka MIDI-viestit menevät laitteesta toiseen tavu kerrallaan järjestyksessä. Asynkronisuus taas tarkoittaa tahdistamattomuutta, eli kohdelaite ei ole tahdistettu ohjauslaitteen kanssa. Kohdelaite siis lukee ja vastaanottaa viestit käyttäen omaa kellotaajuuttaan. Usein käytetty tapa hyödyntää midiä on yhdistää kaksi syntesoijaa toisiinsa, jolloin ne saadaan toimimaan synkronisesti. Painettaessa ensimmäisestä syntesoijasta joitain kosketinta, myös seuraava siihen kytketty syntesoija alkaa soimaan. MIDI:n avulla siis saadaan esim. ohjattua useampaa laitetta samanaikaisesti. Käydään hieman läpi varsinaisen MIDI-viestin rakennetta. (Hirvi & Tuominen 1995, 22.)

MIDI-viesti on perinteisesti 1-3 tavun mittainen digitaalinen viesti(KUVIO 14). Tavuhan sisältää yhteensä kahdeksan bittiä. Näillä kahdeksalla bitillä voidaan ilmaista numeroarvoja väliltä 0-255. MIDI-viestissä ensimmäinen bitti ilmoittaa kohdelaitteelle viestin luonteen. Jäljelle jäävien bittien avulla voidaan vielä ilmaista 128 eri lukuarvoa. Kun käytetään sarjallista tiedonsiirtoa tavut on erotettava toisistaan aloitus- ja lopetusbiteillä, joten yhden kahdeksan bittisen tavun välittämiseen tarvitaan yhteensä 10bittiä. (Hirvi & Tuominen 1995, 22.)



KUVIO 14. Tavun rakenne (Hirvi & Tuominen 1995, 22)

Tarkastellaan hieman tarkemmin yleisintä käytössä olevaa MIDI-viestiä eli Note on-viestiä(KUVIO 15). Kun esim. koskettimistolta painetaan yhtä kosketinta, lähetetään samalla MIDI out-liitännästä kolmen tavun mittainen viesti. Viesti käsittää käskytavun, ensimmäisen arvotavun sekä toisen arvotavun. Käskytavu määrittelee viestin luonteen, esimerkissämme se on NoteOn-viesti. Käskytavun ensimmäisellä neljällä bitillä ilmoitetaan kohdelaitteelle että kyseessä on käskytavu sekä käskytavun osoittama nimi. Neljän viimeisin bitin avulla ilmoitetaan MIDI-kanavan numero (1-6).Ensimmäisellä arvotavulla kerrotaan nuotin numero eli sävelkorkeus, joka voi olla väliltä 0-127. Toisella arvotavulla ilmoitetaan koskettimen painamisen nopeus (velocity), joka voi olla väliltä 0-127. Mitä voimakkaammin koskettimen painallus on tunnistettu niin sitä suuremmalla äänenvoimakkuudella se soitetaan. (Hirvi & Tuominen 1995, 23)



KUVIO 15. MIDI-viestin anatomia (Hirvi & Tuominen 1995, 23)

NoteOn on yksi tavallisimmista viesteistä, mutta niitä on paljon muitakin. Esimerkkinä kun koskettimiston modulaatiopyörää liikautetaan lähetetään samalla kohdelaitteelle sen uutta asentoa vastaava viesti. PitchBend eli sävelkorkeuden ”venytystä” vastaava arvo ja viesti on myös olemassa. Selventääkseni miten PitchBend vaikuttaa niin esimerkiksi MIDI-kitaralla soittaessa kielen venytys lähettää juurikin PitchBend-viestin. Modulaatiopyörä ja pitch bend löytyvätkin melkein jokaisen MIDI-koskettimiston vasemmalta puolelta. Seuraavaksi lista yleisimmistä MIDI-viesteistä(TAULUKKO 1).

TAULUKKO 1. Yleisimmät MIDI-viestit (Hirvi & Tuominen 1995, 24)

Lähetystapa (Ohjauslaite)	Viesti	Vaikutus (Kohdelaite)
Koskettimen painaminen	Note on	Nuotti sytty
Koskettimen vapauttaminen	Note off	Nuotti sammuu
Modulaatiopyörän liikautus	Control Change	Vibrato
Koskettimen painaminen kun se on pohjassa.	After Touch	Vibrato
Soundinvaihtonäppäimen painaminen	Program Change	Käytössä oleva ääni vaihtuu
Pitch benderin (taivutin) liikautus	Pitch Bender Change	Pitch bend

NoteOn välittyy kun painetaan jotain kosketinta. Koskettimet ovat numeroitu 0-127. Esimerkkinä keski-C on numerolla 60.

ControlChange-viestiä voidaan käyttää moniin eri tarkoituksiin. Esimerkissämme modulaatiopyörän käyttö.

AfterTouch eli jälkipaino. Useimmissa MIDI-koskettimistoissa otetaan myös huomioon miten kosketinta painellaan sen jälkeen kun kosketin on painettu pohjaan. Tämän avulla saadaan aikaiseksi esim. tremolo, vibrato tai sävelkorkeuden muutos. Näin voidaan jäljitellä esim. viulun tai intialaisen sitarin vibratoa.

ProgramChange on äänen, efektin tai muun valitun ohjelman muutos.

PitchBend-viesti välittää sävelkorkeuden muutokset esim. jos halutaan simuloida sähkökitaran kielten venytystä.

Kyseiset käskyt ovat kaikki kanavakäskyjä, jotka lähetetään eri MIDI-kanavilla halutuille soittimille. Kanavaviestejä on myös paljon muitakin ja lisäksi käytössä on myös laitteistoviestejä. (Hirvi, Tuominen, 1995, 21-24.)

3.3 General MIDI

Musiikkisekvenssi, joka on sävelletty jollain tietyllä laitteistolla saattaa kuulostaa täysin erilaiselta, kun se yritetään toistaa käyttäen eri laitteistoa. Tietyiltä MIDI-kanavilta ja ääniohjelmilta löytyy todennäköisesti aivan toisenlaiset soitinäänet kuin alkuperäisessä laitteistossa. Vaikka soitinäänet olisivat sattumalta samat niiden käyttäytyminen voi olla hyvin erilaista. On monia tilanteita jolloin musiikkisekvenssin siirrettävyys toiselle laitteistolle ilman ongelmia on tarpeen. Jotta musiikissa voitaisiin käyttää hyödyksi MIDI-ääniä on välttämätöntä, että sävelletty kappale kuulostaa ainakin suunnilleen samalta toistetussa laitteistossa, kuin miten säveltäjä on sen tarkoittanut kuulostavan. Tätä varten kehitettiin General MIDI eli GM.

General MIDI on 1991 hyväksytty laitestandardi, jonka tarkoituksena on yhteinäistää MIDI-käytäntöjä, jolloin sekvenssien siirrettävyys laitteistolta toisella on mahdollisimman helppoa. General MIDI:n avulla määritellään millainen ääni tai soitin on oltava milläkin numerolla. Säveltäjä kappaletta tehdessään tietää että valitessaan kanavan 1 siihen määritetyt sävelet toistuvat kuulijankin laitteistolla akustista pianoa jäljitellen. General MIDI on kuitenkin vain suositus. Monissa kalliimman luokan syntesoijissa tai samplerissa ei käytetä hyödyksi General MIDI:ä. Joissakin laitteistossa on myös General MIDI-äänipankit muitten äänipankkien ohessa, jolloin käyttäjä voi itse valita haluamansa äänipankin. (Hirvi & Tuominen 1995, 91-92.)

General MIDI koostuu seuraavista ennalta määräytyistä osista:

General MIDI-instrumentti. eli syntesoija, äänimoduuli, kortti, kostetinsoitin tai mikä tahansa fyysinen laite johon on tallennettu ääniä General MIDI:n mukaisesti.

Äänikartta. jokaisen 128 määritellyistä ääninumerosta kutsuu esiin tietyn äänen. Jokaisella valmistajalla on vapaus toteuttaa valitut äänet haluamallaan tavalla, mutta ääni numero 36 on aina nauhaton basso, sekä 73 on aina piccolohuilu.

Perkussioäänet. General MIDI:llä määritetään 81 tavallista perkussioääntä. General MIDI-soittimessa niille on annettava taulukon mukaiset nimet ja nuottinumerot. Perkussioäänille varataan aina MIDI-kanava 10.

Muut vaatimukset. Ainakin 24-ääninen polyfonia ja dynaaminen allokaatio. Toisto käyttämällä kuuttatoista MIDI-kanavaa kuudellatoista eri äänellä yhtäaikaisesti. Ainakin 128kpl ennältä määriteltyjä ääninumeroita. Instrumentin on reagoitava painalluksen voimaakkuuteen sekä kanavakohtaiseen jälkipainoon. Instrumentin on reagoitava tiettyihin kontrollereihin luonnollisesti. Mikäli sekvenssi on näiden vaatimusten mukainen ja se tallennettu Standard MIDI File-formaatissa, voidaan se toistaa käyttäen mitä tahansa General Midiä tukevaa laitetta. (Hirvi & Tuominen 1995, 92-94.)

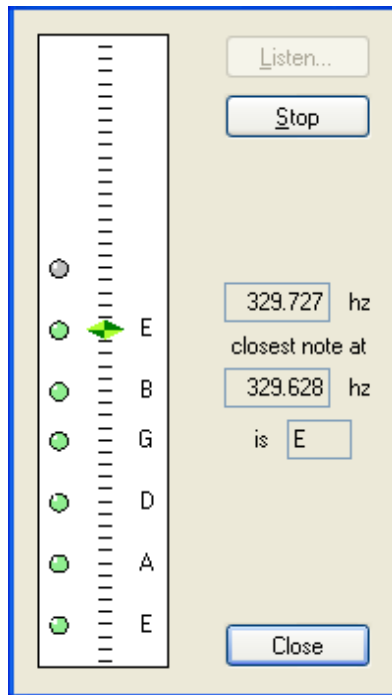
TAULUKKO 3. General MIDI-äänikartta (Hirvi & Tuominen 1995, 93)

General MIDI-äänikartta

Piano	Strings	Synth Lead	Sound Effects
1 Acoustic Grand piano	41 Violin	81 Lead 1 (square)	121 Guitar Fret Noise
2 Bright Acoustic piano	42 Viola	82 Lead 2 (sawtooth)	122 Breath Noise
3 Electric Grand Piano	43 Cello	83 Lead 3 (calliope)	123 Seashore
4 Honky-tonk Piano	44 Contrabass	84 Lead 4 (chiff)	124 Bird Tweet
5 Electric piano 1	45 Tremolo Strings	85 Lead 5 (charang)	125 Telephone Ring
6 Electric piano 2	46 Pizzicato Strings	86 Lead 6 (voice)	126 Helicopter
7 Harpichord	47 Orchestral Harp	87 Lead 7 (fifths)	127 Applause
8 Clavichord	48 Timpani	88 Lead 8 (bass+lead)	128 Gunshot
Chromatic Percussion	Ensemble	Synth pad	
9 Celesta	49 String Ensemble 1	89 Pad 1 (new age)	
10 Glockenspiel	50 String Ensemble 2	90 Pad 2 (warm)	
11 Music Box	51 SynthStrings 1	91 Pad 3 (polysynth)	
12 Vibraphone	52 SynthStrings 2	92 Pad 4 (choir)	
13 Marimba	53 Choir Aahs	93 Pad 5 (bowed)	
14 Xylophone	54 Voice Oohs	94 Pad 6 (Metallic)	
15 Tubular Bells	55 Synth Voice	95 Pad 7 (halo)	
16 Dulcimer	56 Orchestra Hit	96 Pad 8 (sweep)	
Organ	Brass	Synt Effects	
17 Drawbar Organ	57 Trumbet	97 FX 1 (rain)	
18 Percussive Organ	58 Trombone	98 FX 2 (soundtrack)	
19 Rock Organ	59 Tuba	99 FX 3 (crystal)	
20 Church Organ	60 Muted Trumpet	100 FX 4 (atmosphere)	
21 Reed Organ	61 French Horn	101 FX 5 (brightness)	
22 Accordion	62 Brass Section	102 FX 6 (goblins)	
23 Harmonica	63 Synth Brass 1	103 FX 7 (echoes)	
24 Tango Accordion	64 Synth Brass 2	104 FX 8 (sci-fi)	
Guitar	Reed	Ethnic	
25 Acoustic Guitar (nylon)	65 Soprano Sax	105 Sitar	
26 Acoustic Guitar (steel)	66 Alto Sax	106 Banjo	
27 Eletric Guitar (jazz)	67 Tenor Sax	107 Shamisen	
28 Eletric Guitar (clean)	68 Baritone Sax	108 Koto	
29 Eletric Guitar (muted)	69 Oboe	109 Kalimba	
30 Overdriven Guitar	70 English Horn	110 Badpipe	
31 Distortion Guitar	71 Bassoon	111 Fiddle	
32 Guitar Harmonics	72 Clarinet	112 Shanai	
Bass	Pipe	Percussive	
33 Acoustic Bass	73 Piccolo	113 Tinkle Bell	
34 Electric bass (finger)	74 Flute	114 Agogo	
35 Electric bass (pick)	75 Recorder	115 Steel Drums	
36 Fretless Bass	76 Pan Flute	116 Woodblock	
37 Slap bass 1	77 Blown Bottle	117 Taiko Drum	
38 Slap bass 2	78 Shakuhachi	118 Melodic Tom	
39 Synth Bass 1	79 Whistle	119 Synth Drum	
40 Synth Bass 2	80 Ocarina	120 Reverse Symbal	

4 LÖYDETYT OHJELMISTOT

4.1 FFT-guitar tuner



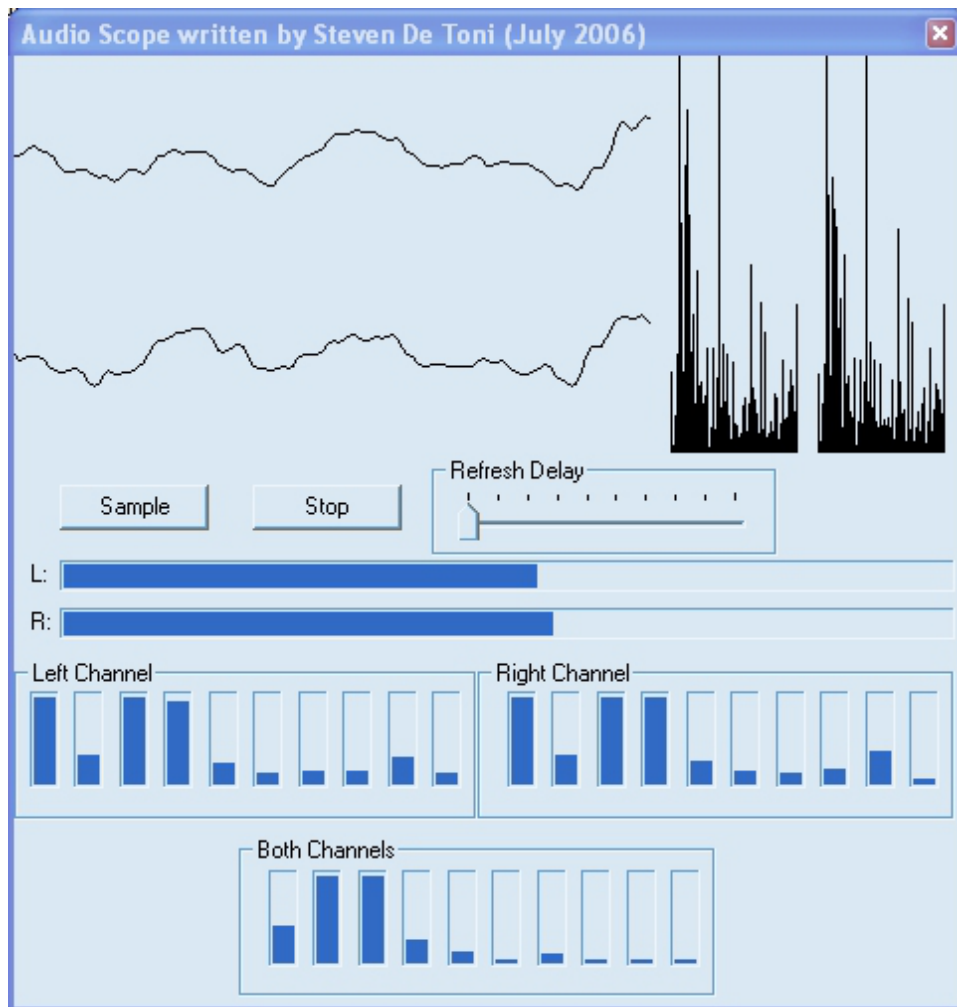
KUVIO 16. Viritysmittari (Notmasteryet 2010)

Ohjelma käyttää hyödyksi Fast Fourier Transform-algoritmia määrittelläkseen kaapatun äänisignaalin perustaajuuden. Ohjelmassa on tätä menetelmää hyödyntäen rakennettu yksinkertainen viritysmittari kitaralle. Ohjelman perusajatus on siis selvittää ja verrata ohjelmalle tulevan audiosignaalin perustaajuutta. Ohjelmasta on saatavilla lähdekoodit, joten kyseistä ohjelmaa voidaan helposti hyödyntää muissakin ohjelmistoprojekteissa.

Ohjelma ja lähdekoodi on saatavilla osoitteessa:

<http://www.codeproject.com/Articles/32172/FFT-Guitar-Tuner>

4.2 Simple Audio Out Oscilloscope and Spectrum Analyzer



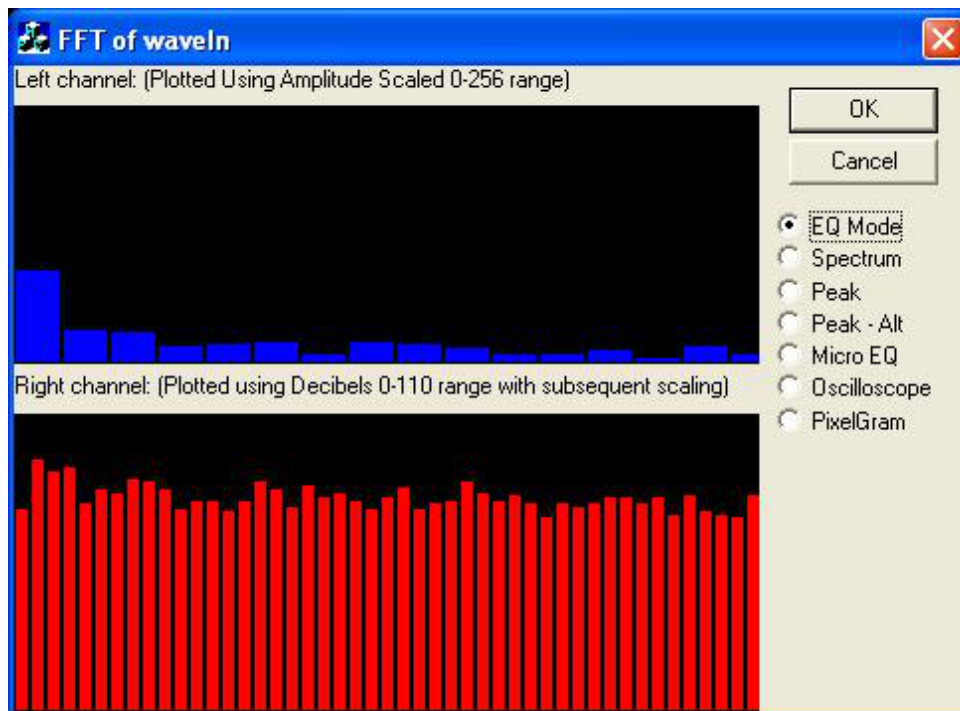
Kuvio 17. Oskilloskooppi (De Toni 2006)

Yksinkertainen sovellus joka auttaa ymmärtämään perusajatuksen miten ääninkäsittelyä voidaan ohjelmallisesti toteuttaa Windows-ympäristössä. Ohjelman avulla voidaan myös tutkia Fast Fourier Transform-algoritmin toimintaa ja toteutusta. Ensin valitaan tietty äänilähde ohjelman tarkasteltavaksi. Ohjelma näytteistää ja tulostaa äänisignaalin aalto- sekä spektrimuodon. Ohjelmasta on saatavilla lähdekoodit, joten kyseistä ohjelmaa voidaan helposti hyödyntää muissakin ohjelmistoprojekteissa.

Ohjelma ja lähdekoodi on saatavilla osoitteessa:

<http://www.codeproject.com/Articles/14873/Simple-Audio-Out-Oscilloscope-and-Spectrum-Analyze>

4.3 FFT of waveIn audio signals



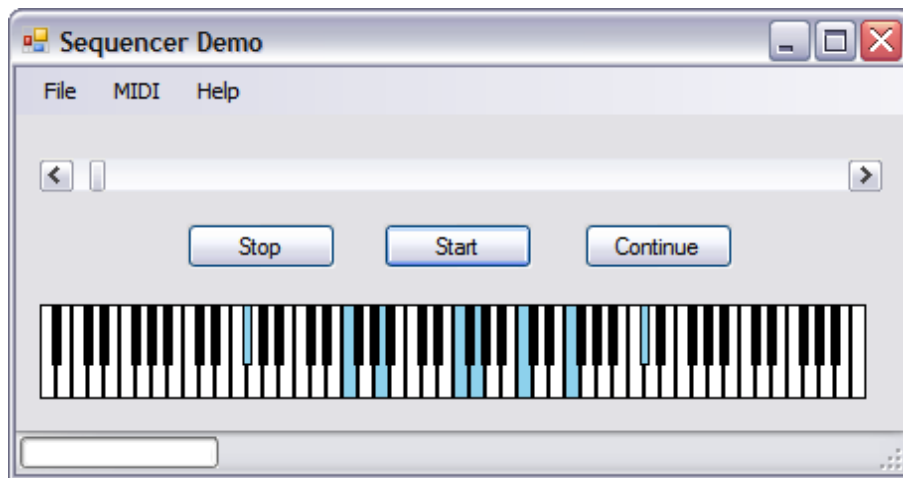
KUVIO 18.FFT of wavein (Ackers 2007)

Hyvin samantapainen ohjelmisto kuin aiemmin esitelty Simple Audio Out Oscilloscope and Spectrum Analyzer. Tämän ohjelmiston avulla voidaan tarkastella äänilähteeseen tulevaa signaalia useamman eri muunnoksen avulla. Ohjelmasta on saatavilla lähdekoodit, joten kyseistä ohjelmaa voidaan helposti hyödyntää muissakin ohjelmistoprojekteissa.

Ohjelma ja lähdekoodi on saatavilla osoitteessa:

<http://www.codeproject.com/Articles/6855/FFT-of-waveIn-audio-signals>

4.4 C# Midi toolkit



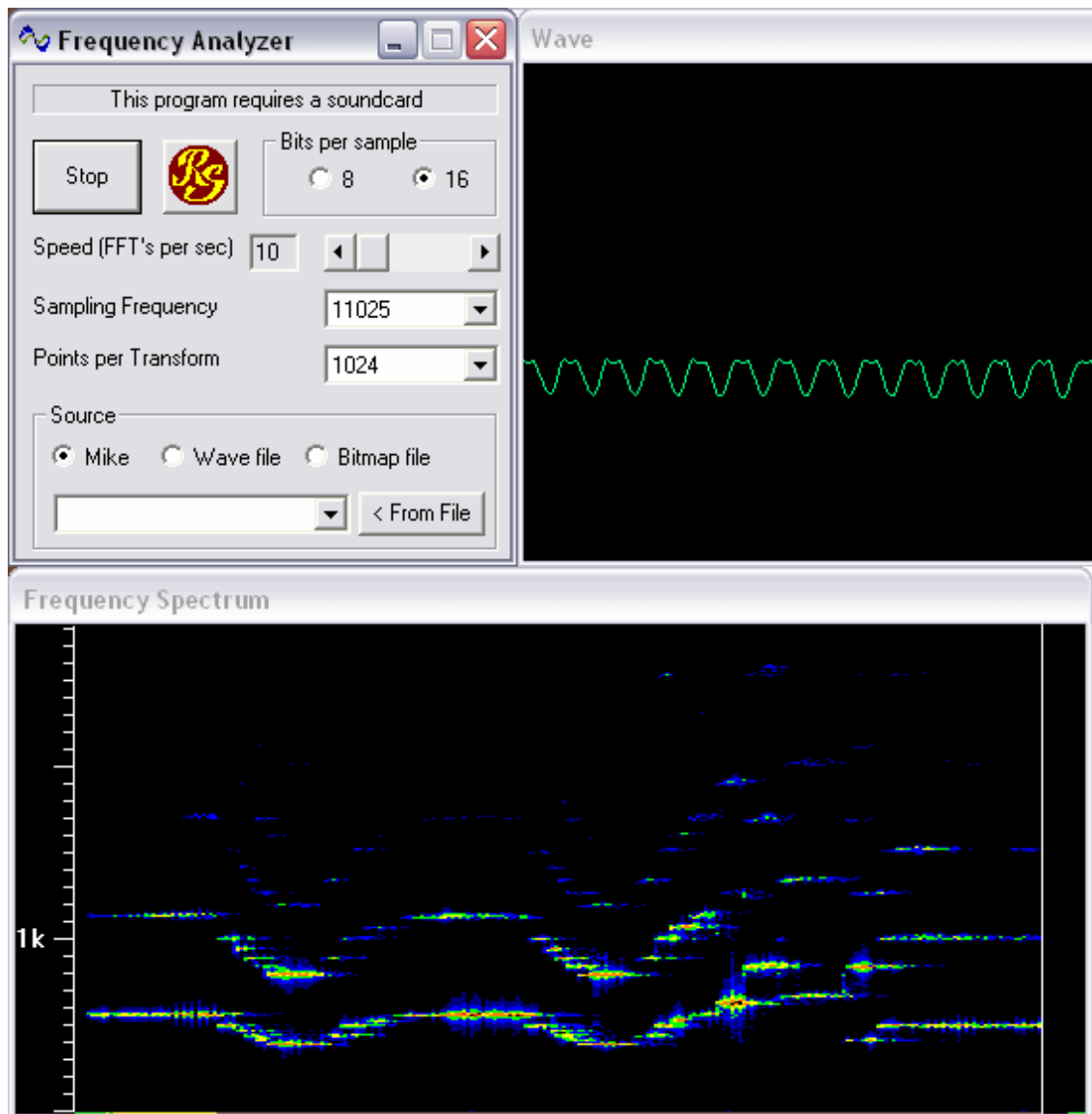
KUVIO 19. C# MIDI toolkit (Sanford 2007)

Midiä hyödyntävä ohjelma jonka avulla voidaan toistaa haluttu MIDI-tiedosto. Ohjelma näyttää soitetut sävelet myös visuaalisesti pianon koskettimistostolla. Midin-avulla voidaan esim. poimia eri sävelet kyseisestä musiikkikappaleesta. Tällöin yhdistämällä sävelen tiedot äänisignaalia tarkkailevaan ohjelmistoon voidaan vertailla onko mikrofonin kautta tuleva signaali sama, kuin MIDI-tiedoston ilmoittama sävel. Toisinsanoen onko soitettu sävel sama mikä se pitäisi olla kyseisessä musiikkikappaleessa. Ohjelmasta on saatavilla lähdekoodit, joten ohjelmaa voidaan helposti hyödyntää muissakin ohjelmistoprojekteissa.

Ohjelma ja lähdekoodi on saatavilla osoitteessa:

<http://www.codeproject.com/Articles/6228/C-MIDI-Toolkit>

4.5 Frequency analyzer



KUVIO 20. Frequency analyser (Reliable software 2006)

Äänen visualisointia monin tavoin. Bittisyvyyttä, näytteenottotaajuutta yms. voidaan muuttaa. Lähdekoodi saatavissa ainakin osittain ohjelman kehittäjän sivuilta.

Ohjelma ja lähdekoodi on saatavilla osoitteessa:

<https://www.relisoft.com/freeware/freq.html>

5 TULOKSET JA RATKAISUEHDOTUKSET

Lopputuloksena löytyi useampikin käyttökelpoinen ohjelmisto, joista myös lähdekooditkin on saatavilla. Sheemu-sovelluksen kannalta operaatiot menevät seuraavasti. Äänisignaali vastaanotetaan mikrofiniliitännän kautta ja muunnetaan analogisesta digitaaliseksi signaaliiksi.

Kaapatun äänisignaalin perustaajuus selvitetään käyttämällä hyödyksi Fourier-muunnoksia ja oikeanlaista näytteistystä. Tämän käsittelyn tulosta voidaan vertailla haluttuun säveleen esittäen myös esim. graafista ohjetta, onko kaapattu äänisignaali oikeassa sävelessä. Midin avulla voidaan poimia eri sävelet kyseisestä musiikkikappaleesta. Tällöin yhdistämällä sävelen tiedot äänisignaalin tietoihin voidaan vertailla onko mikrofonin kautta tuleva signaali sama, kuin MIDI-tiedoston ilmoittama sävel. Toisinsanoen onko soitettu sävel sama mikä se pitäisi olla kyseisessä musiikkikappaleessa.

FFT-guitar tunerin lähdekoodia voidaan käyttää apuna, kun äänisignaalia halutaan vertailla haluttuun säveleen. C# Midi toolkitin avulla saadaan vertailukohdan sävelet ja muita ohjelmia voidaan käyttää avuksi, kun äänisignaalia halutaan esittää halutunlaisella kuvaajalla. Kuvaajan ohjelmoinnissa voidaan käyttää hyödyksi Fourier-muunnoksia ja melkein kaikissa löydetyissä ohjelmissa käytettiin FFT (Fast Fourier Transform) –muunnosta.

6 LÄHTEET

Ackers, F. 2007. FFT of waveIn audio signals. Www-dokumentti. Saatavissa: <http://www.codeproject.com/Articles/6855/FFT-of-waveIn-audio-signals>. Luettu 24.4.2014.

De Toni, S. 2006. Simple Audio Out Oscilloscope and Spectrum Analyzer. Www-dokumentti. Saatavissa: <http://www.codeproject.com/Articles/14873/Simple-Audio-Out-Oscilloscope-and-Spectrum-Analyze>. Luettu 24.4.2014.

Hirvi, J. & Tuominen, A. 1995. Uusi MIDI-kirja. Helsinki: Painatuskeskus Oy.

Karjalainen, M. 2009. Kommunikaatioakustiikka. Pdf-dokumentti. Saatavissa: https://noppa.aalto.fi/noppa/kurssi/s-89.3320/materiaali/S-89_3320_kurssikirja.pdf. Luettu: 24.4.2014.

Notmasteryet. 2010. FFT Guitar Tuner. Www-dokumentti. Saatavissa: <http://www.codeproject.com/Articles/32172/FFT-Guitar-Tuner>. Luettu 24.4.2014.

Peltonen, H. Perkkiö, J. & Vierinen, K. 2012. Insinöörin (AMK) fysiikka osa II. 8.painos. Saarijärvi: Saarijärven OFFSET Oy.

Reliable software. So, how does Frequency Analyzer work. Www-dokumentti. Saatavissa: <https://www.relisoft.com/freeware/freq.html>. Luettu 24.4.2014.

Rumsey, F. & McCormick, T. 1999. Sound and Recording: An Introduction Third edition. Toinen painos. Oxford: Focal Press.

Sanford, L. 2007. C# MIDI Toolkit. Www-dokumentti. Saatavissa: <http://www.codeproject.com/Articles/6228/C-MIDI-Toolkit>. Luettu 24.4.2014.

Smith, S. 1997. The Scientist and Engineer's Guide to Digital Signal Processing. Www-dokumentti. Saatavissa: <http://www.dspguide.com/ch9/1.htm>. Luettu 24.4.2014