



KANDIDAATINTYÖ
**FFT-PERUSTAINEN SPEKTRIANALYYSI USRP-
OHJELMISTORADIOALUSTALLA**

Juha-Mikko Aho

Ohjaaja: Harri Saarnisaari

**ELEKTRONIIKAN JA TIETOLIIKENNETEKNIIKAN
TUTKINTO-OHJELMA**

2020

Aho J-M. (2020) Kandidaatintyö. Oulun yliopisto, Elektroniikan ja tietoliikennetekniikan tutkinto-ohjelma. FFT-perustainen spektrianalyysi USRP-ohjelmistoradioalustalla.

TIIVISTELMÄ

Tässä työssä tutustutaan tehotiheysspektrin estimoinnissa käytettyihin menetelmiin ja käytetään USRP-2900 ohjelmistoradiota spektrianalysointiin. Mittauskaistaksi valittiin WLAN-kaista, mutta työssä käytetyllä menetelmällä voi suorittaa mittauksia myös muilla kaistoilla. Työssä käytetyllä menetelmällä mittaus voidaan tehdä kaistalta, jonka kaistanleveys ylittää laitteen suurimman hetkellisen kaistanleveyden. USRP-2900 laitteen ohjaamiseen ja spektrin analysointia varten laadittiin Matlab-ympäristössä koodi. Koodissa luotiin tehotiheysspektri signaalista FFT-perustaisen Welch-menetelmän avulla. Työssä myös pohdittiin USRP-2900 ohjelmistoradion sopivuutta spektrianalysointiin.

Avainsanat: spektrianalyysi, ohjelmistoradio, USRP-2900, FFT, Welch-menetelmä, Matlab, WLAN.

Aho J-M. (2020) Bachelor's thesis. University of Oulu, Degree Programme in Electronics and Communications Engineering, FFT based spectrum analysis using USRP software defined radio platform.

ABSTRACT

In this work I research information about methods used in power density spectrum estimation and use USRP-2900 software radio to do spectrum analysis. WLAN band was chosen as a measurement band, but other bands can also be measured using this method. Measuring method used in this work can be used to measure a bandwidth which exceeds the maximum transient bandwidth of the measuring equipment. For controlling USRP-2900 and spectrum analysing I wrote a script in Matlab. In the code a power density spectrum was created from the signal using FFT based Welch's method. In this work the suitability of USRP-2900 in spectrum analysis was also evaluated.

Key words: spectrum analysis, software defined radio, USRP-2900, FFT, Welch's method, Matlab, WLAN.

SISÄLLYS

TIIVISTELMÄ	2
ABSTRACT	3
SISÄLLYS.....	4
ALKULAUSE	5
LYHENTEIDEN JA MERKKIEN SELITYKSET	6
1. JOHDANTO	7
2. USRP 2900 JA MATLAB	8
2.1 Ohjelmistoradio	8
2.2 USRP-2900.....	9
2.3 USRP-2900 ja Matlab	9
3. TEORIA.....	12
3.1 Fourier-muunnos	12
3.2 Diskreetti Fourier-muunnos	12
3.3 Nopea Fourier-muunnos.....	13
3.4 Signaalin tehospektrin estimointi, signaalin kohina ja Welch-menetelmä.....	16
4. MITTAUKSET.....	18
4.1 Mitattava WLAN-kaista	18
4.2 Mittaustapa	20
5. YHTEENVETO JA POHDINTA.....	27
6. LÄHTEET	28
7. LIITTEET	30
MATLAB-KOODI	31

ALKULAUSE

Tämä kandidaatintyö on tehty osana elektroniikan ja tietoliikennetekniikan tutkinto-ohjelman kandidaatintutkintoa. Työn aihe oli ohjaajan laatima. Valitsin aiheen, koska olin aikaisemmin Langattoman tietoliikenteen harjoitustyö -kurssilla käyttänyt tässä työssä käytettävää USRP-2900 laitetta Simulink ympäristössä. Tässä työssä Simulink ympäristön sijaan laadittiin Matlabissa koodi, jolla laitetta ohjattiin spektrin analysoinnissa.

Oulussa, toukokuussa 2020.

Juha-Mikko Aho

LYHENTEIDEN JA MERKKIEN SELITYKSET

FPGA	Ohjelmoitava logiikkapiiri
DSP	Digitaalinen signaalinkäsittely
A/D	Analogisesta digitaaliseksi muunnos
D/A	Digitaalisesta analogiseksi muunnos
USRP	Yleiskäyttöinen ohjelmistoradioalusta
DFT	Diskreetti Fourier-muunnos
FFT	Nopea Fourier-muunnos
WLAN	Langaton lähiverkko
WiFi	Standardoitu langaton lähiverkko
RF	Radiotaajuus
DC	Tasajännite
IIR	Jatkuvan impulssivasteen suodatin
$x(t)$	Signaalin amplitudi aikatasossa
$X(f)$	Signaalin amplitudi taajuustasossa
$\theta(f)$	Vaihe
P_{xx}	Periodogrammi
S_{xx}	Jatkuvan satunnaissignaalin tehoteiheys
P_{xx}^w	Welch-menetelmän tehoteiheys
$I(t)$	Radiosignaalin reaalinen osa
$Q(t)$	Radiosignaalin imaginäärinen osa
I/Q	Radiosignaalin kompleksinen esitys
BW	Kaistanleveys
F_s	Näytteenottotaajuus
Hz	Hertsi
dB	Desibeli
t	aika
i	imaginääriluku
f	taajuus

1. JOHDANTO

Tässä työssä käytettiin USRP-2900 laitteen vastaanotinta mitattavan kaistan spektrin analysointiin.

USRP-laitteelle on paljon valmiita esimerkkejä Matlabin Simulink-ympäristössä, jossa voi erilaisia tietoliikenne ratkaisuja simuloida lohkoihin jaettujen osien avulla. Simulinkin lohkoittain rakennettujen esimerkkien aihepiiri keskittyy enimmäkseen erilaisten modulaatioiden havainnollistamiseen sekä bittivirhetodennäköisyyksien laskemiseen.

USRP-laitetta voidaan ohjata myös Matlabin Communications Toolbox -lisäosaan sisältyvien funktioiden avulla. Matlabin esimerkkikirjastossa löytyvät esimerkkikoodit keskittyvät datan lähetykseen ja vastaanottoon USRP-ohjelmistoradiolla. Tässä työssä käytetään Matlabin edellä mainitun lisäosan funktioita signaalin vastaanottamiseen valitulta kaistalta, sekä tehotiheyspektrin laskemiseen ja spektrien kuvitukseen.

Työssä käytetty mittausmenetelmä sisälsi USRP-2900 laitteen suurinta hetkellistä kaistanleveyttä leveämmän mittauskaistan tehotiheyden mittaamista. Koska mittauskaistan leveys ylitti laitteen suurimman hetkellisen kaistanleveyden, täytyi kaistan mittaus suorittaa osissa käyttäen useaa eri keskitaajuutta. Lopuksi eri keskitaajuuksilla mitatut signaalit näytteet koostettiin yhdeksi spektriiksi. Laadittua Matlab-koodia voidaan käyttää mihin tahansa USRP-2900 laitteen suurimman ja pienimmän toiminnallisen taajuuden välisen kaistan mittaamiseen osissa. Tässä työssä kuitenkin valittiin mitattavaksi kaistaksi WLAN-kaista, koska se sisältää paljon liikennettä, joka tekee mittaustuloksista havainnollisemman. Ideaalisessa toteutuksessa USRP-ohjelmistoradiota voitaisiin käyttää reaaliaikaisena pyyhkäisevänä spektrianalysointina edellämainitulla mittausmenetelmällä. Työssä kuitenkin törmättiin reaaliaikaisuutta rajoittaviin tekijöihin.

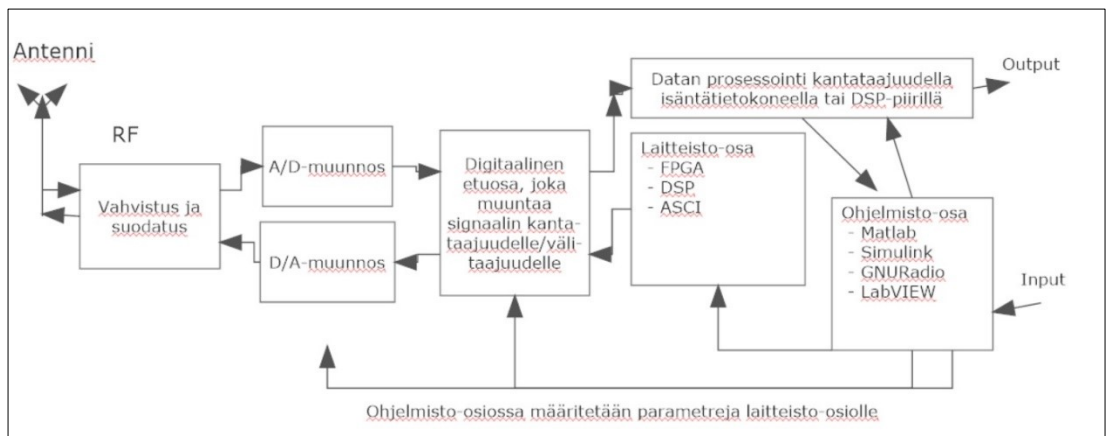
Mitatun signaalit näytteistä laskettiin Matlabissa tehotiheyspektri nopeaan Fourier-muunnokseen perustuvaa Welch-menetelmää käyttäen. Welch-menetelmässä näytteistetty signaali pilkotaan aikatasossa osittain päällekkäin meneviin lohkoihin, joista lasketaan periodogrammit ja lopuksi keskiarvoistetaan pätkittäin luodut periodogrammit yhdeksi spektriiksi. Welch-menetelmän tavoitteena on vähentää signaalin tehotiheyspektrissä näkyvää kohinaista varianssia, ilman että menetelmän aiheuttama näytemäärän desimaatio vähentää taajuusresoluutiota liikaa.

2. USRP 2900 JA MATLAB

Tässä työssä käytettiin USRP-2900 ohjelmistoradiota spektrin analysointiin luomalla Matlab-ympäristössä sitä ohjaava koodi.

2.1 Ohjelmistoradio

Ohjelmistoradiot ovat lähetin-vastaanotinradioalustoja, joissa osa laitteiston fyysisestä toteutuksesta on korvattu ohjelmoitavalla FPGA- tai DSP-piirillä ja ohjelmistoradioon yhteydessä olevalla tietokoneella. Perinteisessä radioalustassa radion toiminnan määrittää sen laitteiston jo valmistusvaiheessa valitut parametrit, joten se ei ole kovin taipuvainen useaan erilaiseen käyttötarkoitukseen. Ohjelmistoradio on halpa ratkaisu radioalustojen suunnittelu- ja testausvaiheessa, koska sillä pystytään toteuttamaan erilaisia laitteistoratkaisuja pelkästään koodia muokkaamalla eikä jokaista testikappaletta tarvitse rakentaa erikseen. Ohjelmistoradioita käytetään myös moderneissa tietoliikennetarkoituksissa, joissa radiolta vaaditaan monimutkaista digitaalista signaalin prosessointia tai jopa tekoälyn kaltaisia ominaisuuksia tietoliikennekaistan käytön optimoimiseen. [1, 2]



Kuva 1. Lohkokaavio ohjelmistoradioissa käytettävästä periaatteesta.

Ohjelmistoradion rakenne voidaan jakaa kuvassa 1 näkyvän lohkokaaavion tapaan laitteisto- ja ohjelmisto-osioihin (*engl. hardware and software*). Laitteisto-osaan kuuluu antennin lisäksi radiotaajuuksilla ja välitaajuuksilla toimivat suodattimet, vahvistimet sekä analogia-digitaali- ja digitaali-analogiamuuntimet. Laitteisto-osa koostuu fyysisistä adaptiivisista komponenteista, joita ohjataan asettamalla niihin tietyt parametrit ohjelmisto-osiossa. Ohjelmistoradioilta vaaditaan erittäin laajaa toimintakykyä erilaisilla taajuusalueilla ja laajaa dynaamista aluetta, tästä johtuen laitteisto-osan toiminta-alueen täytyy olla todella laaja. Ohjelmistoradion toiminnallisen taajuus- ja dynamiikka-alueen määrittääkin laitteisto-osa. Hetkellisen kaistanleveyden suuruuteen ohjelmistoradiossa vaikuttaa laitteen maksimaalinen näytteenottoaika. [1, 3]

Laitteisto- ja ohjelmisto-osan rajapinta on A/D- ja D/A- muuntimet. Vastaanottimessa antenniin saapunut signaali muutetaan A/D-muuntimen kohdalla analogisesta signaalista digitaaliseen muotoon biteiksi niin sanotulle kantataajuudelle (*engl. baseband*). Tässä muodossa olevalle signaalille voidaan tietokoneella tai itse

ohjelmistoradion FPGA- ja DSP-piireissä suorittaa erilaisia signaalinprosessointimenetelmiä; esimerkiksi virheenkorjauskoodausta, demodulointia ja modulointia, spektrianalyysiä ja digitaalista suodatusta. Lähettimessä rajapintana toimii D/A-muunnin, jossa signaali muutetaan analogiseksi signaaliksi laitteisto-osaa varten. [1, 3]

2.2 USRP-2900

USRP eli yleiskäyttöinen ohjelmistoradioalusta (*engl. Universal Software Radio Peripheral*) on National Instrumentsin tytäryhtiön Ettus Researchin tuottama ohjelmistoradioiden tuoteryhmä joka on suunniteltu suhteellisen halvaksi vaihtoehdoksi opiskelu-, harrastus- ja tutkimuskäyttöön. USRP radioalustojen käyttöä varten Ettus Research on julkaissut avoimen lähdekoodin GNU Radio -ohjelmiston, joka tarjoaa erilaisia työkaluja USRP-radioilla toteutettaviin projekteihin. GNU radio perustuu C++ ja Python ohjelmointikieliin. [1, 2, 3]

Tässä työssä käytetään USRP-2900 mallin ohjelmistoradiota. USRP-2900 sisältää myös lähettimen, mutta sitä ei tehospektrin tutkimisessa tarvita. Taulukossa 1 on USRP-2900 laitteelle asetettavien parametrien raja-arvot [2, 4]. Näistä kaikista oleellisin tässä työssä on suurin hetkellinen kaistanleveys, jota ei voi asettaa 56 MHz suuremmaksi.

Taulukko 1. USRP-2900 laitteen ominaisuudet.

Taajuusalue	70 MHz – 6 GHz
Taajuuden säätötarkkuus	< 1 kHz
Vahvistusalue	76 dB
Vahvistuksen säätötarkkuus	1,0 dB
Taajuusherkkyyys	2,5 ppm
Suurin hetkellinen kaistanleveys	56 MHz
Kohinakerroin	5 dB – 7 dB
DAC	12 bittiä
Suurin I/Q näytenopeus	Jatkuva: $15 \cdot 10^6$ näytettä sekunnissa Hetkellinen: $61,44 \cdot 10^6$ näytettä sekunnissa
Suurin tuloteho	-15 dBm

USRP-2900 käyttää tietokoneen kanssa USB 3.0 tai USB 2.0 yhteyttä. Jos käytetään USB 2.0 yhteyttä, pitää USRP:hen kytkeä myös virtalähde. USB-yhteyttä varten tietokoneelle täytyi asentaa ajuri. Ajurin asentamiseen käytettiin Zadig-ohjelmistoa, joka tunnistaa automaattisesti USB-yhteydessä olevat laitteet sekä etsii ja asentaa niille sopivat ajurit. [1]

2.3 USRP-2900 ja Matlab

Matlab käyttää USRP:n kanssa kommunikointiin Communications Toolbox -lisäosaa, joka sisältää erilaisia tietoliikennejärjestelmien fyysisen tason (*engl. physical layer*) simulointiin käytettäviä työkaluja. Communications Toolbox myös sisältää Ettus Researchin USRP-radioiden käyttämiseen tarkoitettua tukipaketin. [5]

Matlab ja sen edellämainittu lisäosa sisältää useita valmiiksi laadittuja komentokooodeja eli funktioita joilla USRP:tä voidaan ohjata ja sen kanssa kommunikoida. Ensimmäinen funktio jota USRP:tä käyttäessä tarvitaan on siihen yhdistämiseen käytetty *findsdru* [6]. Se etsii ja palauttaa tietokoneeseen yhdistetyn USRP:n tilaviestin Matlabin konsoliin. Tilaviestissä näkyy USRP:n IP-osoite tai sarjanumero, mallin tyyppi ja laitteen status, joka on “*Success*” kun laite on valmis kommunikoidaan isäntätietokoneen kanssa tai “*Busy*” kun USRP suorittaa laitteeseen annettua komentoa eikä ole valmis kommunikoidaan isäntätietokoneen kanssa. Kuvassa 2 näkyy esimerkki käytetystä *findsdru* komennosta ja sen lopputuloksesta.

```
>> findsdru
Checking radio connections...
ans =

struct with fields:

    Platform: 'B200'
    IPAddress: ''
    SerialNum: '30C2BF8'
    Status: 'Success'
```

Kuva 2. Kuvassa on esimerkki *findsdru* komennosta ja sen lopputuloksesta.

Kun toimiva yhteys on varmistettu, laitteen kanssa kommunikointiin ja vastaanotinantennin datan tallentamista varten USRP alustetaan komennolla *comm.SDRUReceiver*. Tässä komennossa alkuosa “*comm*” tarkoittaa, että komento on Communication Toolbox lisäosan komento ja “*SDRUReceiver*” tarkoittaa ohjelmistoradiovastaanotinta (*engl. software defined radio unit receiver*). Komentoon voidaan sisältää useita eri muuttujia, jotka määrittävät esimerkiksi vastaanottimen keskitaajuuden, näytteenottotaajuuden, otettujen näytteiden määrän ja signaalin vahvistuksen. [6]

```
radio = comm.SDRUReceiver('Platform', 'B200', 'SerialNum', '30C2BF8',
'CenterFrequency', 2.401e9, 'MasterClockRate', 32e6, 'SamplesPerFrame', 1024,
'OutputDataType', 'double','Gain',60);
```

Kuva 3. Esimerkki käytetystä *comm.SDRUReceiver* funktiosta.

Kuvassa 3 näkyy esimerkki *comm.SDRUReceiver* komennosta. Kuvan 3 komennossa vastaanotettu data tallennetaan *radio* -nimiseen muuttujaan. Komennossa on asetettu laite vastaanottamaan keskitaajuudella 2,401 GHz, näytteenottotaajuudella 32 MHz, ottamaan 1024 näytettä joiden bittimääräinen pituus on ‘*double*’ eli 64 bittiä ja signaalin vahvistukseksi 60 desibeliä.

Taulukossa 1 näkyy USRP-2900 laitteisto-osan asettamat rajoitukset vastaanottimelle keskitaajuuden, vahvistuksen ja kaistanleveyden suhteen. Esimerkiksi maksimaalinen keskitaajuus, joka voidaan vastaanottaa on 6 GHz.

Vastaanotettu data on I/Q-signaalia, joka on kompleksinen lukuarvo. I/Q-signaalissa radiosignaali

$$s(t) = I(t) + i * Q(t) \quad (2.3.1)$$

on jaettu reaaliosaan

$$I(t) = \text{real}(s(t)) \quad (2.3.2)$$

ja sen kanssa 90° vaihe-erossa olevaan imaginääriosaan [7]

$$Q(t) = \text{imag}(s(t)). \quad (2.3.3)$$

I/Q-signaali on siis vektori, jonka elementit ovat radiosignaalin reaali-osa $I(t)$ ja imaginääriosa $Q(t)$ [7].

Laitteen lähettämä ja Matlabin vastaanottaama datasiignaali tallennetaan vektoriin, jonka pituus määritettiin komennossa muuttujassa "*Samples per frame*" [6]. Jos näytteenotossa käytetään integrointiaikaa, näytevektoreita otetaan useita ja ne tallennetaan matriisiin. Integrointiaika määrittää sen kuinka kauan näytevektoreita, eli näyteikkunoita otetaan.

3. TEORIA

Signaalin tehosppektri on taajuuden funktio, joten aikatasossa mitattu signaali täytyy muuntaa taajuuden funktioksi seuraavin menetelmin.

3.1 Fourier-muunnos

Fourier-muunnos on matemaattinen menetelmä, jolla jaksollinen ajan funktio $g(t)$ voidaan esittää eri taajuuksien sinisignaalien summana $G(f)$. Fourier-muunnoksella on myös vastaoperaatio, jolla eri taajuuksien sinisignaalien spektri voidaan esittää ajan funktiona. [8]

Fourier-muunnoksen ja sen vastaoperaation yhtälöt ovat

$$G(f) = \int_{-\infty}^{\infty} g(t)e^{-i2\pi t f} dt \quad \text{ja} \quad (3.1.1)$$

$$g(t) = \int_{-\infty}^{\infty} G(f)e^{i2\pi t f} df. \quad (3.1.2)$$

Yhtälön (3.1.1) operaatiota kutsutaan funktion $g(t)$ Fourier-muunnokseksi, jossa signaali esitetään taajuusavaruudessa ja yhtälön (3.1.2) operaatio on Fourier-muunnoksen vastaoperaatio, jossa signaalin esitys siirretään reaali-aikatasoon ajan funktioksi. Fourier-muunnokseen liittyy myös amplitudi [8]

$$|G(f)| = \sqrt{\text{Real}(f)^2 + \text{Imag}(f)^2} \quad (3.1.3)$$

ja vaihe

$$\phi(f) = \arctan\left(\frac{\text{Imag}(f)}{\text{Real}(f)}\right). \quad (3.1.4)$$

3.2 Diskreetti Fourier-muunnos

Diskreetti Fourier-muunnos on nimensä mukaisesti aikadiskreetin signaalin muunnos. Jatkuva-aikaisessa Fourier-muunnoksessa, joka näkyy yhtälössä (3.1.1), muunnos tehdään integroimalla ajan funktion $g(t)$ yli, eli käytännössä se vaatisi äärettömän määrän näytteitä. Ääretön määrä näytteitä ei ole signaalinmittauksessa kuitenkaan mahdollista. Diskreetissä Fourier-muunnoksessa ajan funktio $g(t)$ on jaettu diskreetteihin arvoihin x_1, x_2, \dots, x_N , jotka vastaavat näytteenottohetkiä t_1, t_2, \dots, t_N , jossa $t_2=t_1+\Delta t$ ja $t_N=t_1+(N-1)\Delta t$. Vakio Δt on näytteenotossa käytetty näytteenottoväli ja N on otettujen näytteiden määrä. [9]

Diskreetin Fourier-muunnoksen ja sen vastaoperaation yhtälöt ovat

$$X(k) = \sum_{n=0}^{N-1} x(n)e^{-i2\pi kn/N}, \quad n = 0, \dots, N-1 \quad \text{ja} \quad (3.2.1)$$

$$x(n) = \frac{1}{N} \sum_{k=0}^{N-1} X(k)e^{i2\pi kn/N}, \quad k = 0, \dots, N-1. \quad (3.2.2)$$

Diskreetti Fourier-muunnos perustuu Eulerin kaavaan [9]

$$X(k) = \sum_{n=0}^{N-1} x(n) e^{-\frac{i2\pi kn}{N}} = \sum_{n=0}^{N-1} x(n) \left[\cos\left(\frac{2\pi}{N}kn\right) - i \sin\left(\frac{2\pi}{N}kn\right) \right]. \quad (3.2.3)$$

Yhtälössä (3.2.1) on diskreetti Fourier-muunnos ja yhtälössä (3.2.2) on sen käänteismuunnos. Diskreetti Fourier-muunnos siis koostuu siniaaltojen summasta kuten jatkuva-aikainen Fourier-muunnos, mutta sen taajuuskomponentit ovat jakautuneet taajuuksille $X(k)$, $k=0, \dots, N-1$, jossa $X(k)$ vastaa todellista taajuutta

$$f = \frac{k}{N\Delta t}, \quad (3.2.4)$$

jolloin suurin taajuuskomponentti on $X(k=N-1)$ ja $X(k=0)$ vastaa nollataajuutta eli esimerkiksi jännitteen mittauksessa tasajännitekomponenttia. Yhtälöstä (3.2.4) laskettu suurin taajuuskomponentti määräytyy Nyquistin näytteettotaajuuden mukaan

$$f_{Nyq} = \frac{1}{2\Delta t}, \quad (3.2.5)$$

joka on todellinen suurin mitattava taajuus, josta saadaan informaatiota näytevälillä Δt . Tarkasteltaessa Fourier-muunnetun signaalin spektriä, huomataankin että taajuuskomponentit itseasiassa peilautuvat näytteenottotaajuuden f_s molemmin puolin, eli näytemäärällä $N=8$ peilautuvat taajuuskomponentit $X(1)=X(7)$, $X(2)=X(6)$ ja niin edelleen. Signaalin mittauksessa jokainen $f_{Nyq} = \frac{1}{2\Delta t}$ suurempi taajuuskomponentti aiheuttaa laskostumista, joka näkyy taajuusspektrissä peilitaajuuksien siirtymisenä peilattavan taajuuskaistan päälle. Tästä aiheutuu aikatasossa signaalin vääristymistä, jos Nyquistin taajuutta suurempia taajuuskomponentteja ei suodateta pois. [9]

3.3 Nopea Fourier-muunnos

Diskreetti Fourier-muunnos sisältää useita kompleksilukujen yhteen- ja kertolaskuja, jotka vaativat suuren määrän laskentatehoa pienilläkin näytemäärillä. Jotta reaaliaikainen spektrin analysointi olisi edes jotenkin mahdollista, täytyy näytemäärää laskea tai muunnoksen laskutapaa muuttaa. Tähän ratkaisuna on kehitetty laskumenetelmä, jossa muunnoksessa olevien kompleksikertoimien symmetrisuutta hyödynnetään asettamalla näytemäärä esimerkiksi luvun kaksi potenssin kokonaislukua vastaavaksi. On myös olemassa muita versioita kuten ”radix4”, jossa kantalukuna on luvun kaksi sijaan luku neljä [10]. Tällä laskumenetelmällä kompleksinen kerroin onnistutaan muuttamaan useassa tapauksessa joko arvoon 0, -1 tai 1. Seuraava menetelmä on E. O. Birghamin ja R.E. Morrowin laatiman esimerkin mukainen [9].

Oletetaan, että signaalista otetaan $N=4$ näytteitä ja siitä lasketaan diskreetti Fourier-muunnos. Lisäksi merkitään yhtälössä (3.2.1) olevaa eksponenttikerrointa termillä $W^{nk} = e^{-i2\pi kn/N}$. Saadaan matriisi [9]

$$\begin{pmatrix} X(0) \\ X(1) \\ X(2) \\ X(3) \end{pmatrix} = \begin{pmatrix} W^0 & W^0 & W^0 & W^0 \\ W^0 & W^1 & W^2 & W^3 \\ W^0 & W^2 & W^4 & W^6 \\ W^0 & W^3 & W^6 & W^9 \end{pmatrix} \begin{pmatrix} x_0(0) \\ x_1(1) \\ x_2(2) \\ x_3(3) \end{pmatrix}, \quad (3.3.1)$$

josta huomataan, että diskreetin Fourier-muunnoksen laskemiseen vaadittu yhteen- ja kertolaskuoperaatioiden määrä on näytteiden määrän neliö N^2 [9]. Oletetaan myös, että muunnos tehdään tietokoneella, joka kykenee tekemään yhden laskuoperaation yhden nanosekunnin aikana. Taulukossa 2 on laskettu eri näytemäärillä tarvittujen laskuoperaatioiden määrä ja niiden vaatima aika, kun tietokone laskee yhden laskuoperaation yhden nanosekunnin aikana. Huomataan, että muunnoksen laskemiseen tarvittava aika kasvaa liian suureksi jo suhteellisen pienillä näytemäärillä.

Taulukko 2. DFT vaatima laskuoperaatioiden määrä.

Näytteiden määrä	Laskuoperaatioiden määrä	Laskenta-aika
4	16	16 ns
40	1 600	1,6 μ s
400	160 000	160 μ s
4 000	16 000 000	16 ms
40 000	1 600 000 000	1,6 s

Huomataan, että jos halutaan ottaa uusi näytevektori esimerkiksi 1,6 μ s välein, täytyy näytemäärän olla alle 40. Laskuoperaatioiden määrän vähentämiseen ja sitä myöten laskenta-ajan lyhenemiseen ja sallitun näytemäärän kasvuun on kehitetty nopea Fourier-muunnos. Nopea Fourier-muunnos hyödyntää kompleksisen kertoimen $W^{nk} = e^{-i2\pi nk/N}$ symmetrisuutta [9]

$$W^{nk} = W^{nk \bmod N}. \quad (3.3.2)$$

Yhtälö (3.3.2) edellyttää, että näytteenotossa näytemääräksi valitaan $N = 2^\gamma$, jossa γ on kokonaisluku. Sovelletaan nopeaa Fourier-muunnosta yhtälössä (3.3.1) olevaan matriisiin sijoittamalla siihen (3.3.2), jolloin $N=4=2^2$. Matriisi (3.3.2) kääntyy muotoon [9]

$$\begin{aligned} \begin{pmatrix} X(0) \\ X(1) \\ X(2) \\ X(3) \end{pmatrix} &= \begin{pmatrix} W^{0 \bmod 4} & W^{0 \bmod 4} & W^{0 \bmod 4} & W^{0 \bmod 4} \\ W^{0 \bmod 4} & W^{1 \bmod 4} & W^{2 \bmod 4} & W^{3 \bmod 4} \\ W^{0 \bmod 4} & W^{2 \bmod 4} & W^{4 \bmod 4} & W^{6 \bmod 4} \\ W^{0 \bmod 4} & W^{3 \bmod 4} & W^{6 \bmod 4} & W^{9 \bmod 4} \end{pmatrix} \begin{pmatrix} x_0(0) \\ x_1(1) \\ x_2(2) \\ x_3(3) \end{pmatrix} \\ &= \begin{pmatrix} W^0 & W^0 & W^0 & W^0 \\ W^0 & W^1 & W^2 & W^3 \\ W^0 & W^2 & W^0 & W^2 \\ W^0 & W^3 & W^2 & W^1 \end{pmatrix} \begin{pmatrix} x_0(0) \\ x_1(1) \\ x_2(2) \\ x_3(3) \end{pmatrix}. \end{aligned} \quad (3.3.3)$$

Seuraavaksi matriisi muunnetaan muotoon [9]

$$\begin{pmatrix} X(0) \\ X(2) \\ X(1) \\ X(3) \end{pmatrix} = \begin{pmatrix} 1 & W^0 & 0 & 0 \\ 1 & W^2 & 0 & 0 \\ 0 & 0 & 1 & W^1 \\ 0 & 0 & 1 & W^3 \end{pmatrix} \begin{pmatrix} 1 & 0 & W^0 & 0 \\ 0 & 1 & 0 & W^2 \\ 1 & 0 & W^2 & 0 \\ 0 & 1 & 0 & W^2 \end{pmatrix} \begin{pmatrix} x_0(0) \\ x_0(1) \\ x_0(2) \\ x_0(3) \end{pmatrix}. \quad (3.3.4)$$

Lasketaan yhtälön oikeanpuoleisten matriisien tulo, josta saadaan [9]

$$\begin{pmatrix} x_1(0) \\ x_1(1) \\ x_1(2) \\ x_1(3) \end{pmatrix} = \begin{pmatrix} 1 & 0 & W^0 & 0 \\ 0 & 1 & 0 & W^2 \\ 1 & 0 & W^2 & 0 \\ 0 & 1 & 0 & W^2 \end{pmatrix} \begin{pmatrix} x_0(0) \\ x_0(1) \\ x_0(2) \\ x_0(3) \end{pmatrix}. \quad (3.3.5)$$

Huomataan, että $x_1(0)$ ja $x_1(1)$ laskemiseen tarvitaan molempiin yksi kompleksinen kertolasku ja yhteenlasku, koska $W^0 = 1$ [9]. Rivin $x_1(2)$ laskemiseen tarvitaan vain yksi kompleksinen yhteenlasku, koska $W^0 = -W^2$. Joten vektorin $X_1(k)$ laskemiseen tarvitaan yhteensä neljä yhteenlaskua ja kaksi kertolaskua [9]. Seuraavaksi sijoitetaan vektori $X_1(k)$ matriisiin (3.3.4) ja lasketaan $X(k)$ [9].

$$\begin{pmatrix} X(0) \\ X(2) \\ X(1) \\ X(3) \end{pmatrix} = \begin{pmatrix} 1 & W^0 & 0 & 0 \\ 1 & W^2 & 0 & 0 \\ 0 & 0 & 1 & W^1 \\ 0 & 0 & 1 & W^3 \end{pmatrix} \begin{pmatrix} x_1(0) \\ x_1(1) \\ x_1(2) \\ x_1(3) \end{pmatrix}, \quad (3.3.6)$$

jossa $X(0)$ saadaan laskettua yhdellä kertolaskulla ja yhteenlaskulla, $X(2)$ yhdellä yhteenlaskulla, $X(1)$ yhdellä kompleksisella kertolaskulla ja yhteenlaskulla ja $X(3)$ vain yhdellä yhteenlaskulla [9]. Joten vektori

$$\overline{X(k)} = \begin{pmatrix} X(0) \\ X(2) \\ X(1) \\ X(3) \end{pmatrix} \quad (3.3.7)$$

saatiin laskettua yhteensä $\frac{N\gamma}{2} = 4$ kompleksisella kertolaskulla ja $N\gamma = 8$ kompleksisella yhteenlaskulla. [9]

Suurella näytämäärän N arvolla FFT vaatima laskuoperaatioiden määrä noudattaa kaavaa $N \log_2 N$, kun taas DFT kaavaa N^2 [9]. Kun näytteiden määrää kasvatetaan, huomataan DFT ja FFT vaadittavien laskuoperaatioiden suhdeluvun merkittävää laskua. Taulukossa 3 näkyy DFT ja FFT -muunnoksiin vaadittavien laskuoperaatioiden määrän ero [9].

Taulukko 3. FFT ja DFT vaadittavien laskuoperaatioiden määrä.

Näytteiden määrä	DFT laskuoperaatioiden määrä	FFT laskuoperaatioiden määrä	Suhdeluku
4	16	8	0,50
40	1 600	212	0,13
400	160 000	3458	0,04
4000	16 000 000	47863	0,03

Kun näytemäärää kasvatetaan, FFT ja DFT vaatimien laskuoperaatioiden määrän suhdeluku lähenee arvoa 0,01 [9].

Suuremmalla kuin $N=4$ näytemäärillä FFT toteutetaan samaan tapaan kuin yllä, mutta lasku muunnetaan useaan vastaavaan vaiheeseen [9].

3.4 Signaalin tehosppektrin estimointi, signaalin kohina ja Welch-menetelmä

Signaalin tehosppektrin estimointi tarkoittaa arviointia satunnaissignaalin tehosta taajuuden funktiona. Kun tutkitaan vastaanotettua signaalia, siihen ei kuitenkaan sisälly pelkästään vastaanotettu signaali, vaan myös siihen summautuva kohina. Kohina on satunnainen muuttuja signaalin tehossa, jota ei voi kokonaan poistaa, mutta sitä voi vähentää ja suodattaa tilastollisin menetelmin. Yleensä kohinan aikakeskiarvo on nolla, mutta sillä on nollasta poikkeava varianssi. Kohina ei muuta itse signaalia, mutta se vaikuttaa vastaanotettuun kokonaisuuteen. [11]

Kohina voi olla jatkuvaa kuten esimerkiksi lämpökohina, tai hetkellistä kuten esimerkiksi muiden laitteiden lähettämien signaalien aiheuttamat häiriöt. Hetkellisiä häiriöitä pystyy suodattamaan kasvattamalla niin sanottua integrointiaikaa, jos se on mahdollista. Integrointiajan kasvattaminen tarkoittaa näytejonojen ottamista pitkältä ajalta ja näytejonojen keskiarvoistamista. [11]

Periodogrammi on satunnaisen ja aikadiskreetin eli näytteistetyn signaalin tehon spektrin estimointi. Periodogrammin kaava on [12]

$$P_{xx}(f) = \frac{1}{M} |\sum_{k=0}^{M-1} X(k)|^2, \quad (3.4.1)$$

jossa M on näytemäärä. Valkoisen kohinan periodogrammin varianssi käyttäytyy todellisen tehotehosppektrin tehon neliön mukaisesti [14 s.22]

$$\text{Var}\{P_{xx}(f)\} \rightarrow S_{xx}^2(f). \quad (3.4.2)$$

Vaikka näytemäärää M nostettaisiin kohti ääretöntä, sillä ei ole vaikutusta periodogrammin varianssiin, eli periodogrammin heikkous on sen varianssin aiheuttama kohina. [12, 13, 14 s.22]

Welch-menetelmä on keino vähentää tehon spektrin estimoinnista aiheutuvaa varianssia. Welch-menetelmässä signaali $x(n)$ pilkotaan aikatasossa L pituisiin K määrään segmentteihin $x_j(n)$ jotka menevät osittain päällekkäin [15]

$$x_1(j) = x(j), \text{ jossaa } j = 0, \dots, L - 1 \quad (3.4.3)$$

$$x_2(j) = x(j + D), \text{ jossaa } j = 0, \dots, L - 1 \quad (3.4.4)$$

$$\dots$$

$$x_K(j) = x(j + (K - 1)D), \text{ jossaa } j = 0, \dots, L - 1. \quad (3.4.5)$$

Seuraavaksi jokainen L pituinen segmentti ikkunoidaan ja niille lasketaan uusi periodogrammi muuntamalla ikkunoidut segmentit nopealla Fourier-muunnoksella

$$\bar{P}_{xx}^j(f) = \frac{1}{MU} \left| \sum_{n=0}^{M-1} x_j(n) w(n) e^{-i2\pi f n} \right|^2, j = 0, 1, \dots, L - 1, \quad (3.4.6)$$

jossa U on normalisointifunktio ikkunointifunktio w(n):n teholle [15]. Normalisointifunktion kaava on [15]

$$U = \frac{1}{M} \sum_{n=0}^{M-1} w^2(n). \quad (3.4.7)$$

Lopuksi Welch-menetelmällä saatu tehotiheyspektri saadaan keskiarvoistamalla ikkunoidut päällekkäiset periodogrammit [15]

$$P_{xx}^W(f) = \frac{1}{L} \sum_{n=0}^{L-1} \bar{P}_{xx}^n(f). \quad (3.4.8)$$

Eli Welch-menetelmällä signaalin kohinaisuutta pyritään vähentämään ikkunoimalla se aikatasossa ja ottamalla ikkunoista lasketun periodogrammin keskiarvo. Welch-menetelmässä oikean pituisten ikkunointien ja ikkunoiden päällekkäisyyden määrällä voidaan vaikuttaa siihen kuinka paljon signaalin varianssia saadaan vähennettyä. Menetelmästä saadulla hyödyllä on kuitenkin varjopuolena se, että taajuusresoluutio heikkenee. Oikein käytettynä Welch-menetelmällä saadaan varianssi pidettyä kurissa ilman että taajuusresoluutio menee liian pieneksi. [15]

Jos signaalissa on kohinaa jonka aikakeskiarvo ei ole nolla, se näkyy FFT-muunnatussa signaalissa nollataajuuden ympäristössä olevana tehona. Matemaattinen menetelmä tämän tyyppisen kohinan suodattamiseen on signaalin aikakeskiarvon vähentäminen signaalista, niin sanottu DC-suodatus. [8]

4. MITTAUKSET

Mittaukset suoritettiin ohjaamalla USRP-2900 -laitetta Matlabissa laaditulla koodilla, joka löytyy liitteenä. Koodin avulla laitteella voidaan ottaa haluttu määrä näytteitä valitulta kaistalta ja koostaa tallennetuista näytteistä tehotiheyspektri. Koodissa käytettiin apuna Mathworks-sivulta löytyvää esimerkkiä ehdollisen funktion toteutukseen [16].

Liitteessä 1 löytyvässä koodissa voidaan valita mittauskaistan alin ja ylin taajuus sekä kuinka monella eri keskitaajuudella mittaus suoritetaan. Mittauksen hetkellinen keskitaajuus ja näytteenottonopeus riippuu koko valitun mittauskaistan leveydestä ja käytettyjen keskitaajuuksien määrästä. Taulukossa 4 on listattu mittauksessa käytetyt parametrit.

Taulukko 4. Mittauksessa käytetyt parametrit.

Parametri	Valittu arvo
Alin mitattu taajuus	2401 MHz
Ylin mitattu taajuus	2483 MHz
Käytettyjen keskitaajuuksien määrä	14
Hetkellinen kaistanleveys	5,86 MHz
Näytemäärä per ikkuna	1024
Integrointiaika	2 s
Vastaanottimen vahvistus	60 dB
Näytteenottotaajuus	11,72 MHz
Desimaatioluku	512

Taulukossa 4 mainittu desimaatioluku on RF-taajuudella otettujen näytteiden määrän suhde nollataajuudelle muutettujen näytteiden määrään. Kun nollataajuudella otettujen näytteiden määrä on 1024, laitteen oletusasetuksena käytetty desimaatioluku 512 aiheuttaa sen, että jokaista 1024 näytteen ikkunaa kohden otetaankin $1024 \cdot 512 = 524\,288$ näytettä ennen desimointia. Laite käyttää desimointia, eli datan näytemäärän supistamista, muuntaessaan signaalin RF-taajuudelta välitaajuuden kautta nollataajuudelle sekä signaalin suodattamiseen [6, 17 s. 28]. Muunnos tapahtuu laitteen AD9364-mallin RF-osassa, jossa signaalia käsitellään esimerkiksi desimaatiosuodattimien avulla [17 s. 28].

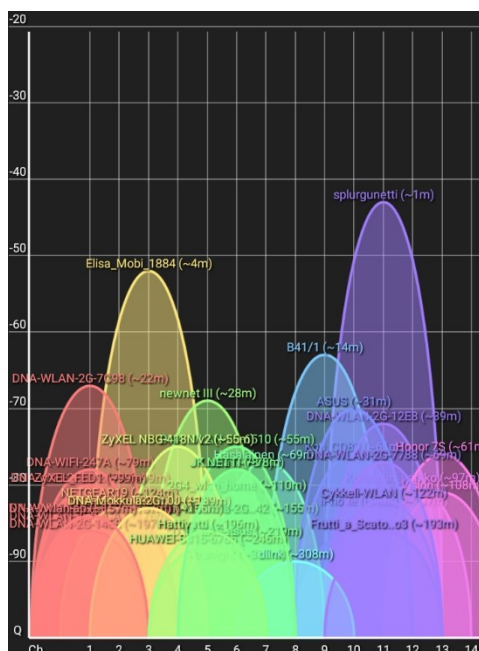
4.1 Mitattava WLAN-kaista

Tässä työssä mitattiin kotitalouksissa langattomissa reitittimissä käytetyn IEEE 802.11 -standardin WLAN-kaistan tehospektriä. Kaista ulottuu taajuuksille 2,401 – 2,495 GHz ja se on jaettu yhteensä neljääntoista kanavaan, joiden kolmentoista ensimmäisen kanavan keskitaajuuksien väli on 5 MHz ja viimeisen kanavan keskitaajuus erottuu edellisestä 12 MHz. Kanavien kaistanleveys on 22 MHz. Kanavat menevät siis osittain päällekkäin. Tässä työssä mitattiin 13 ensimmäistä kanavaa. Taulukossa 5 on listattu WLAN-kanavien taajuusjako [18].

Taulukko 5. 2.4 GHz WLAN-kanavan taajuudet.

Kanava	Alin taajuus, MHz	Keskitaajuus, MHz	Ylin taajuus, MHz
1	2401	2412	2423
2	2406	2417	2428
3	2411	2422	2433
4	2416	2427	2438
5	2421	2432	2443
6	2426	2437	2448
7	2431	2442	2453
8	2436	2447	2458
9	2441	2452	2463
10	2446	2457	2468
11	2451	2462	2473
12	2456	2467	2478
13	2461	2472	2483
14	2473	2484	2495

Kuvassa 4 näkyvästä älypuhelimella WiFi Analyzer -sovelluksella otetusta tehospektristä näkee tämän taajuusjaon käytännössä eri reitittimien välillä. Sovellus käyttää älypuhelimien WiFi-vastaanotinta spektrianalysaattorina. Käytän tätä spektriä myös verrokkina USRP-radiolla otetulle spektrille koodin ja mittaustavan toimivuuden analysointiin.



Kuva 4. Älypuhelimien WiFi Analyzer -sovelluksella otettu WLAN-kaistan tehospektri.

4.2 Mittaustapa

Liitteen 1 Matlab-koodissa voidaan valita taajuusväli, jolla mittaus tehdään, sekä se kuinka monella hetkellisellä keskitaajuudella mittaus tehdään. USRP-2900 suurin hetkellinen kaistanleveys on 56 MHz ja jos halutaan esimerkiksi mitata n. 100 MHz leveää 802.11b standardin WLAN-kaistaa, täytyy mittauksessa kaista pilkkoa vähintään kahdelle eri hetkelliselle keskitaajuudelle. Tässä työssä tehdyssä mittauksessa mitattava kaista on 2401 – 2483 MHz ja se on pilkottu 14 eri hetkellisellä keskitaajuudella otettuun pätkään. Mittauksessa käytettävä hetkellinen kaistanleveys on siis

$$BW = \frac{2483 \text{ MHz} - 2401 \text{ MHz}}{14} = 5,857 \text{ MHz}, \quad (4.2.1)$$

joka tarkoittaa sitä että pienin mahdollinen käytettävä näytteenottonopeus Nyquistin näytteenottoteoreeman mukaan on mitattavan kaistan leveys eli 5,857 MHz. Tämä johtuu siitä, että mitattava kaista sijoittuu kantataajuudelle muunnoksen jälkeen nollataajuuden ympärille kaksipuoleisesti siten, että ylin taajuus on taajuusakselin kohdassa $BW/2$ ja taajuusakselin negatiivisen puolen ääripää sijoittuu kohtaan $-BW/2$. Eli näytteenottotaajuus $F_s = BW$ ei aiheuta laskostumista [19 s. 38-42]. Mittauksessa valittu näytteistystaajuus on kuitenkin kaksinkertainen pienimpään vaadittavaan näytteistystaajuuteen nähden

$$F_s = 2 * BW = 11,714 \text{ MHz}. \quad (4.2.2)$$

USRP-laite täytyy alustaa jokaiselle uudelle hetkelliselle keskitaajuudelle mittauksen aloittamiseksi. Tämä on mittauksessa käytetyn koodin ajamisessa eniten aikaa vievä kohta. Jokaiselle uudelle keskitaajuudelle vaihtaessa laite prosessoi käskyä yli 10 sekuntia. Tästä johtuen laite ei pysty mittaamaan kyseistä spektriä reaaliajassa vaan koko kaistan mittaus kestää useita minuutteja.

Koodi ottaa jokaisella keskitaajuudella 1024 näytteen vektoreita, eli ikkunoita, kahden sekunnin ajan ja tallentaa ne matriisiin

$$\begin{pmatrix} \text{Näyte nr. 1 ensimmäinen ikkuna} & \cdots & \text{Näyte nr. 1 viimeinen ikkuna} \\ \vdots & \ddots & \vdots \\ \text{Näyte nr. 1024 ensimmäinen ikkuna} & \cdots & \text{Näyte nr. 1024 viimeinen ikkuna} \end{pmatrix}. \quad (4.2.3)$$

Näyteikkunoiden ottamiseen käytetty kaksi sekuntia on niin sanottu integrointiaika. Integrointiaika määrää mitattujen ikkunoiden määrän. Kahden sekunnin aikana ikkunoita saatiin desimoinnin jälkeen 157 kappaletta per keskitaajuus. Yhden näyteikkunan ottamiseen aikaa kului 0,0128 s.

Koodissa luodaan lopussa spektri vektorista (4.2.4), jossa on näyteikkunat keskiarvoistettu yhdeksi vektoriksi. Keskiarvoistettujen näytteiden vektori on muotoa

$$\begin{pmatrix} \text{Näyte nr. 1 keskiarvoistettu} \\ \vdots \\ \text{Näyte nr. 1024 keskiarvoistettu} \end{pmatrix}. \quad (4.2.4)$$

Jokaisella keskitaajuudella generoidaan kolme spektrikuvaajaa. Ensimmäinen kuvaaja on yhdestä 1024 näytteen ikkunasta, jossa on pelkästään tehty Welch-

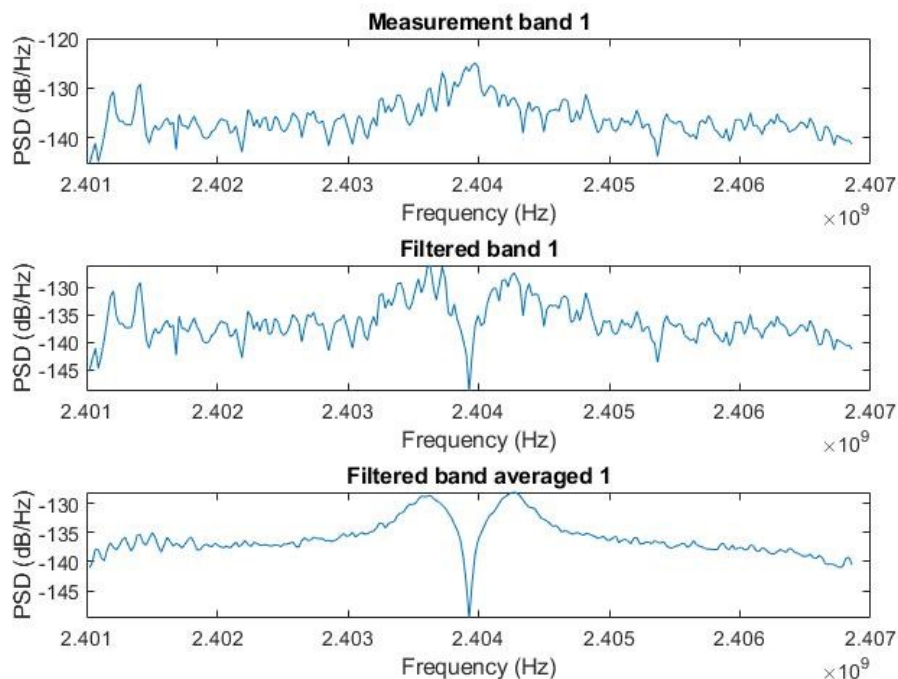
menetelmällä tehotiheysspektrin estimointi. Toinen kuvaaja on myös vain yhdestä ikkunasta Welch-menetelmällä estimoitu spektri, mutta siitä on poistettu nollataajuuden ympäristössä oleva DC-komponentti jatkuvan impulssivasteen suodattimella (*engl. infinite impulse response filter*) eli IIR-suodattimella ennen spektrin estimointia. Kolmannessa kuvaajassa on kaikkien keskitaajuudella otettujen ikkunoiden keskiarvoistettu Welch-menetelmällä estimoitu tehospektri, josta on myös ennen spektrin estimointia poistettu DC-komponentti.

Liitteessä 1 löytyvässä koodissa Welch-menetelmää käytettiin Matlabissa valmiina löytyvällä ”*pwelch()*” funktiolla [20]. Funktiossa käytettiin muuttujina signaalin näytevektoria, näytteistystaajuutta ja saadun spektrin muodoksi valittiin ”*centered*”, eli kaksipuoleinen spektri. Saadun spektrin taajuudet sijoittuivat välille

$$-\frac{F_s}{2} + \text{keskitaajuus} < f < \frac{F_s}{2} + \text{keskitaajuus}, \quad (4.2.5)$$

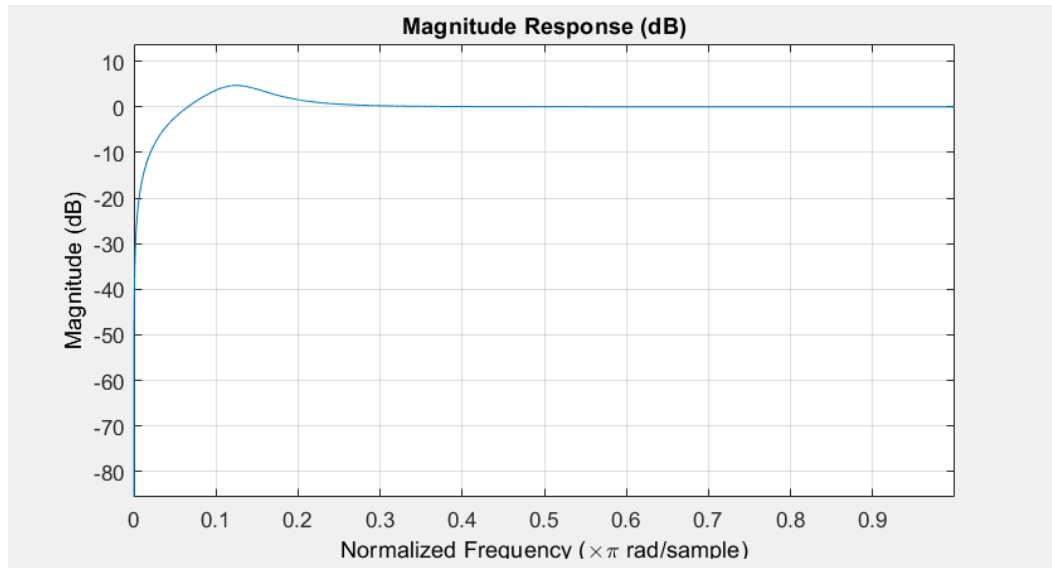
jossa F_s on näytteenottotaajuus ja keskitaajuus on jokaisen mittauspätkän hetkellinen keskitaajuus.

Koodin lopussa kaksi spektrikuvaajaa luodaan lisäämällä jokaiselta näytteenottotaajuudelta aiemmin luodut Welch-metodilla estimoidut tehospektrit peräkkäin yhdeksi koostetuksi spektriksi. Ensimmäisessä spektrikuvaajassa signaaleista ei ole poistettu DC-komponenttia, eikä ikkunoita ole keskiarvoistettu, vaan spektrissä näkyy pelkästään yksi ikkuna. Toisessa koostetussa spektrissä on poistettu DC-komponentti ja kaikki ikkunat on keskiarvoistettu. Keskiarvoistus näkyy spektrissä varianssin vähentymisenä.



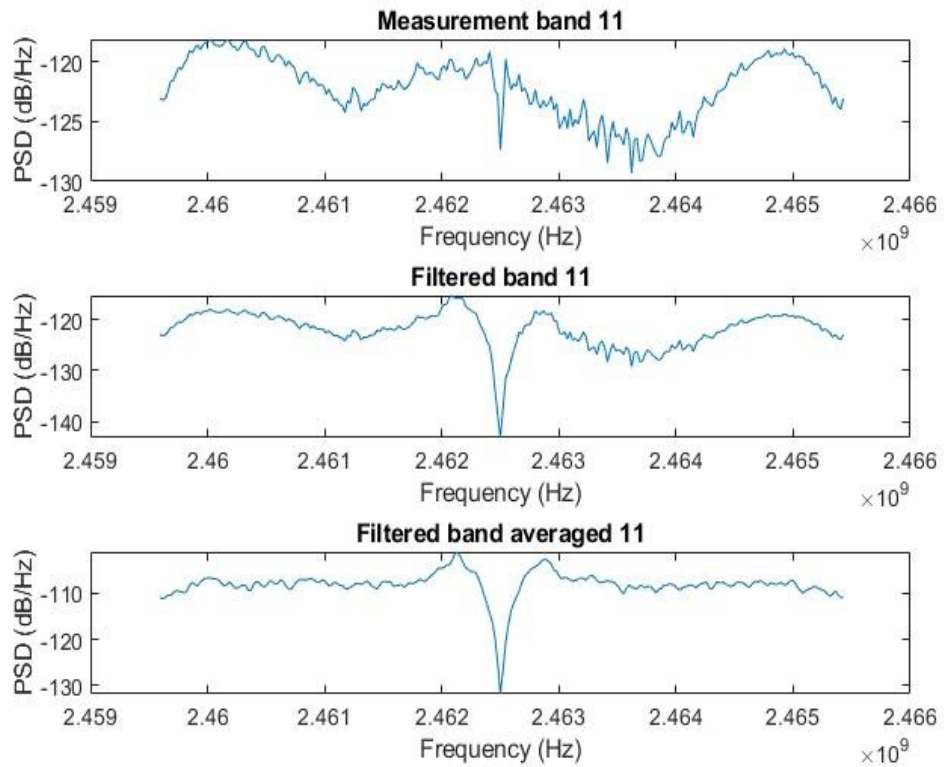
Kuva 5. Ensimmäisellä hetkellisellä keskitaajuudella aikaisemmin kuvatuilla tavoilla otetut tehotiheysspektrit.

Kuvassa 5 on kolme spektriä ensimmäiseltä keskitaajuudelta. Koko mitattava kaista oli 2401 MHz – 2483 MHz ja se oli jaettu 14 osaan. Joten yllä olevassa kuvassa hetkellinen kaistanleveys on n. 5,86 MHz, keskitaajuus 2403,93 MHz ja näytteenottotaajuus 11,7MHz. Ensimmäisessä Welch-menetelmällä luodussa tehon estimoinnissa näkyy nollataajuuden ympärillä oleva pääkeila. Se johtuu aikaisemmin mainitusta FFT-muunnoksen aiheuttamasta $X(0)$ -taajuuskomponenttiin kerääntyvästä signaalin aikakeskiarvosta eli DC-komponentista, sekä ikkunoinnissa käytetystä Hamming-siirtofunktion muodosta. Keskimmäisessä spektrissä siitä on pyritty pääsemään eroon IIR-suodattimesta luodulla DC-suodattimella. Alimmassa spektrissä kaikki kahden sekunnin aikana mitatut 1024 näytteen ikkunat on keskiarvoistettu.



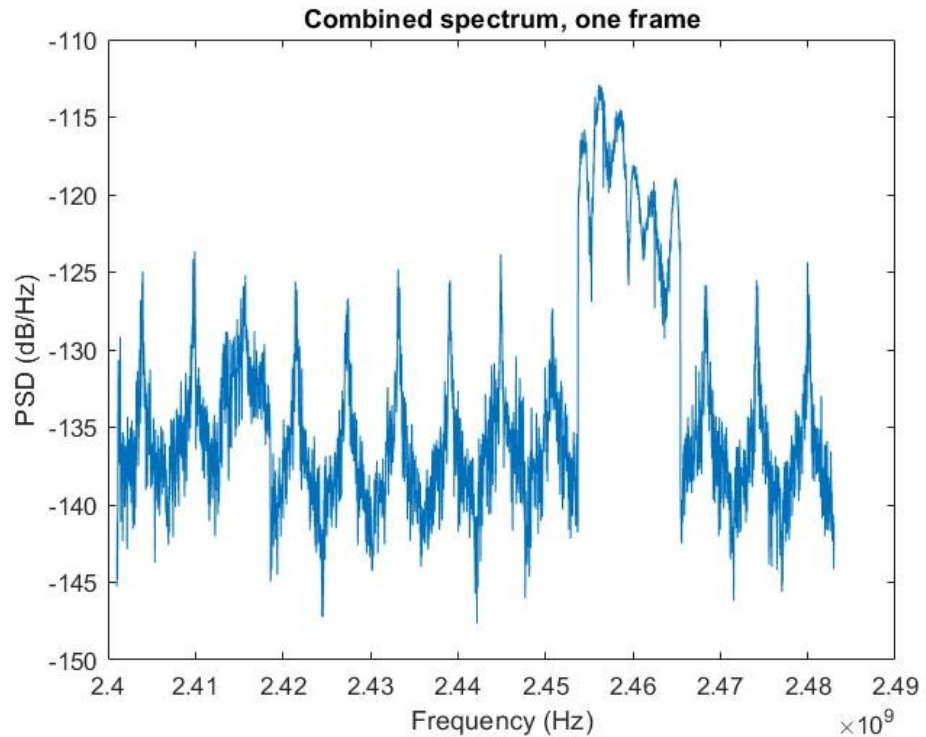
Kuva 6. DC-komponentin suodatuksessa käytetty IIR-suodattimen normalisoitu taajuusvaste.

Kuvassa 6 on DC-suodattimena käytetyn IIR-suodattimen normalisoitu amplitudivaste. Kuten kuvasta huomaa, se suodattaa nollataajuuden ympäristössä signaalia. IIR-suodattimen asteluku on 3 ja normalisoitu päästökaistan kulmataajuus $0.1 \cdot 2\pi$ rad/sample.



Kuva 7. Yhdennellätoista mitatulla keskitaajuudella otetut spektrit.

Kuvassa 7 näkyy 11. käytetyltä keskitaajuudelta mitattu spektri, jossa tehotiheys on -110 dB/Hz seudulla. Kun spektriä vertaa 1. keskitaajuudella mitattuun spektriin, huomataan että tehotiheys on selvästi suurempi. Siitä voisi päätellä, että tällä kaistalla on muutakin kuin pelkkää kohinaa. Eli tällä taajuudella on todennäköisesti WLAN liikennettä.



Kuva 8. Koostettu tehotiheyspektri eri keskitaajuuksilla otetuista näytteistä. Tässä spektrissä tehon estimointi on suoritettu vain yhdestä ikkunasta, eli aikaintegrointia ei ole suoritettu. Myöskään DC-komponentteja ei ole suodatettu.

Kuvassa 8 näkyy koostettu tehospektri, jossa kaikilla keskitaajuuksilla otetut Welch-menetelmällä estimoidut spektrit on lyöty peräkkäin. Spektrissä on DC-piikkien aiheuttamia epämuodostumia, jotka häiritsevät spektrin tulkintaa. Spektrissä näkyy myös paljon kohinan aiheuttamaa varianssia.

Kun kuvassa 9 näkyvää mitattua spektriä vertaa kuvassa 10 olevaan älypuhelimien WiFi Analyzer -sovelluksella saatuun spektriin, voidaan todeta että mittaus on aika lähellä todellisuutta.

5. YHTEENVETO JA POHDINTA

Työssä tutustuttiin signaalin tehosiheyden spektrin laskemiseen liittyviin menetelmiin ja tehtiin koemittaus laaditulla Matlab-koodilla WLAN-kaistalta.

Koodi on toimiva ratkaisu USRP-laitteen käyttöön laajennetun kaistan spektrianalyysissä, mutta sillä ei kuitenkaan pystytä tekemään reaaliaikaista mittausta. Suurin reaaliaikaisuutta rajoittava tekijä oli USRP:n asettaminen uudelle keskitaajuudelle, joka kesti yli kymmenen sekuntia jokaisen keskitaajuuden kohdalla. Eli toteutusta voidaan käyttää pitkän ajan kuluessa kaistalla tapahtuvien ilmiöiden tutkimiseen.

Mittauksessa on myös käytössä useita erilaisia parametrejä: otettujen näytteiden määrä, hetkellisten keksitaajuuksien määrä mittauksessa, integrointiaika, desimaatioluku, Welch-menetelmässä käytettyyn ikkunointiin liittyvät valinnat ja DC-suodatuksen vaste. Tässä työssä tehdyssä mittauksessa näiden parametrien optimaaliseen säätämiseen ei kovin paljoa kiinnitetty huomiota, mutta tarvittaessa niitä säätämällä mittaustuloksista saisi vielä paljon parempia. Esimerkiksi useamman hetkellisen keskitaajuuden käyttö nostaisi mittauksen resoluutiota sillä tavoin, että eri kaistoilla tapahtuvan liikenteen erottaisi toisistaan paremmin. Kuitenkin hetkellisten keskitaajuuksien määrän nostaminen myös kasvattaa mittausaikaa.

Myös Welch-menetelmässä käytettyjen parametrien optimointi auttaisi saamaan spektristä paremman. Welchin-menetelmässä käytettyjen parametrien avulla voitaisiin tasapainotella spektriin aiheutuvan varianssin ja taajuusresoluution menetyksen välillä.

Ideaali DC-suodatin suodattaisi FFT-muunnoksen aiheuttaman signaalin aikakeskiarvon kerääntymisen nollataajuuden ympärille ilman, että se aiheuttaa saaduissa spektreissä nähtyä liian voimakasta suodattumista, joka näkyy nollataajuuksien tienoilla olevina kuoppina.

Kuvassa 9 näkyvästä spektristä myös huomataan, kuinka tehosiheyden jakaantuminen ei ole tasainen eri keskitaajuuksilla käytettyjen mittausten rajamailla, vaan pätkittäin mitattujen lohkojen välillä on suuri pudotus tehosiheydessä. Todellisuudessa näillä taajuuksilla ei ole tällaista tehon pudotusta, vaan se on mittaustavan aiheuttama. Ehkä tähänkin löytyy parannuskeino.

Ehkä myös mittauksessa kohdattuun reaaliaikaisuusongelmaan saattaa löytyä ratkaisu. Jos laitteen uudelle keskitaajuudelle asettamisen aiheuttama viive pystytään eliminoimaan, pystytään laitetta käyttämään reaaliaikaisena spektrianalysoijana. Tämä on suurin ongelma liittyen laitteen käyttämiseen spektrianalysoijana.

6. LÄHTEET

- [1] Kärkkäinen K. (2020) Langattoman tietoliikenteen harjoitustyö: USRP-2900-ohjelmistoradio, luentomateriaali.
- [2] Liuski T. (2019) AM- ja DSB-modulaatioiden toteuttaminen Simulink-ohjelmistolla USRP-ohjelmistoradioalustalle, kandidaatintyö.
- [3] National Instruments (luettu 20.4.2020) What is NI USRP Hardware, URL: <http://www.ni.com/white-paper/12985/en/>
- [4] National Instruments (luettu 20.4.2020) USRP-2900 datasheet, URL: <http://www.ni.com/fi-fi/shop/select/software-defined-radio-device>
- [5] Mathworks (luettu 12.5.2020) Communication toolbox, URL: <https://se.mathworks.com/products/communications.html>
- [6] Mathworks (luettu 12.5.2020) Comm.SDRuReceiver dokumentaatio, URL: <https://se.mathworks.com/help/supportpkg/usrpradio/ug/comm.sdrureceiver-system-object.html>
- [7] National Instruments (luettu 20.4.2020) What is I/Q data?, URL: <http://www.ni.com/tutorial/4805/en/>
- [8] Saarela J. (2020) Elektroninen mittaustekniikka: Spectrum analyzer, luentomateriaali.
- [9] Brigham, E. O., & Morrow, R. E. (1967). The fast fourier transform. IEEE Spectrum, 4(12), 63-70. doi:10.1109/MSPEC.1967.5217220
- [10] DSP related.com (luettu 12.5.2020) Free books: Mathematics of the DFT: Radix2 FFT, URL: https://www.dsprelated.com/freebooks/mdft/Radix_2_FFT.html
- [11] Saarela J. (2020) Elektroninen mittaustekniikka: System and noise, luentomateriaali
- [12] Mathworks (luettu 12.5.2020) Periodogram, URL: <https://se.mathworks.com/help/signal/ref/periodogram.html>
- [13] Mathworks (luettu 12.5.2020) Bias and variability in the periodogram, URL: <https://se.mathworks.com/help/signal/ug/bias-and-variability-in-the-periodogram.html>
- [14] Schuster G. (2010) (luettu 2.6.2020), Spectrum estimation using periodogram, Bartlett and Welch, URL: http://www.laurent-duval.eu/Documents-Common/Schuster_G_2010_lect_spectrum_upbw.pdf
- [15] Welch P. (1967). The fast Fourier transform for the estimation of power spectra: a method based on time averaging over short, modified periodograms. IEEE Transactions on audio and electroacoustics, vol. AU-15, no. 2, pp. 70-73

- [16] Mathworks (luettu 12.5.2020) Apply Conditional Execution dokumentaatio, URL: <https://se.mathworks.com/help/supportpkg/usrpradio/ug/apply-conditional-execution.html>
- [17] Analog Devices (luettu 2.6.2020), RF Agile Transceiver AD9364 datasheet, URL: <https://www.analog.com/media/en/technical-documentation/data-sheets/AD9364.pdf>
- [18] Electronics notes (luettu 20.4.2020) Wi-Fi Channels, Frequencies, Bands & Bandwidths URL: <https://www.electronics-notes.com/articles/connectivity/wifi-ieee-802-11/channels-frequencies-bands-bandwidth.php>
- [19] Valkama M. (luettu 2.6.2020), Tampereen teknillinen yliopisto, Complex-valued signals and systems: basic principles and applications to radio communications and radio signal processing, luentomateriaali, URL: <http://www.cs.tut.fi/kurssit/TLT-9707/presentations/Complex-Signals-and-Radios-short-2pp.pdf>
- [20] Mathworks (luettu 12.5.2020) Pwelch, URL: <https://se.mathworks.com/help/signal/ref/pwelch.html>

7. LIITTEET

Liite 1 Matlab-koodi

MATLAB-KOODI

```
clear all; close all; clc;

minFrequency = 2401e6; %alin mitattava taajuus
maxFrequency = 2483e6; %ylin mitattava taajuus
numberOfBands = 14; %kuinka monella keskitaajuudella mittaus

BW = (maxFrequency - minFrequency)/numberOfBands;
% mittauksen hetkellinen kaistanleveys
% USRP-2900 suurin hetkellinen kaistanleveys on 56MHz,
% joten maksimi sallittu BW=56MHz/2, kun Fs=2*BW
%Jos näytteenottotaajuus on minimi eli Fs=1*BW, niin BWmax=56MHz

masterClockRate = 2*BW; %näytteenottotaajuus eli Fs, minimi on 1*BW

sampleRate = masterClockRate/512; %desimoinnin aiheuttama pudotus
frameTime = 1024/sampleRate;
%Yhden 1024 näytteen ikkunan ottamiseen käytetty aika

teho = zeros();
taajuus = zeros();
kaikkinaytteet = zeros();

for i= 0: +1: numberOfBands-1

centerFrequency = minFrequency + BW/2 + BW*i;
% Hetkellinen keskitaajuus

radio = comm.SDRuReceiver('Platform', 'B200', 'SerialNum', '30C2BF8',
'CenterFrequency', centerFrequency, 'MasterClockRate', masterClockRate,
'SamplesPerFrame', 1024, 'OutputDataType', 'double','Gain',60);
radioList = findsdru();
dcblokkaus = dsp.DCBlocker('Order',3,'NormalizedBandwidth', 0.1,'Length',512)

if strcmp(radioList.Status, 'Success')
% Otetaan näyteframeja 2 sekunnin ajan
timeCounter = 0;
while timeCounter < 2
[data, len] = radio();
if len > 0
% Oikeaa dataa saatiin laitteesta
y = dcblokkaus(data);
if kaikkinaytteet == 0
kaikkinaytteet = y;
else
kaikkinaytteet = [kaikkinaytteet,y];
end
end
```

```

        yksframe = data;
        % Päivitetään kello
        timeCounter = timeCounter + frameTime;
    end
end
len = 0;
%plotataan yksi frame per keskitäajuus
[pxx,f] = pwelch(yksframe,[],[],[],masterClockRate,'centered');
f=f/2+centerFrequency;
figure(i+1)
subplot(3,1,1)
plot(f, 10*log10(pxx))
title(['Measurement band ',num2str(i+1)])
xlabel('Frequency (Hz)')
ylabel('PSD (dB/Hz)')

%plotataan dc-blokattu yksi frame
[pyy,f] = pwelch(y,[],[],[],masterClockRate,'centered');
f = f/2 + centerFrequency;
subplot(3,1,2)
plot(f, 10*log10(pyy))
title(['Filtered band ', num2str(i+1)])
xlabel('Frequency (Hz)')
ylabel('PSD (dB/Hz)')

%otetaan pwelch kaikista ja keskiarvoistetaan framet
[pzz,f] = pwelch(kaikkinaytteet,[],[],[],masterClockRate,'centered');
pzzavg = mean(pzz,2);
f = f/2 + centerFrequency;
subplot(3,1,3)
plot(f, 10*log10(pzzavg))
title(['Filtered band averaged ', num2str(i+1)])
xlabel('Frequency (Hz)')
ylabel('PSD (dB/Hz)')

kaikkinaytteet = 0;

if i == 0
    teho = pxx;
    tehoavg = pzzavg;
    taajuus = f;
else
    teho = [teho; pxx];
    tehoavg = [tehoavg; pzzavg];
    taajuus = [taajuus; f];
end
end
release(radio);

```



```
release(dcblokkaus);
end
%plotataan yhden framen spektri
figure(i+2)
plot(taajuus, 10*log10(teho))
title('Combined spectrum, one frame')
xlabel('Frequency (Hz)')
ylabel('PSD (dB/Hz)')

%plotataan kaikkien framejen spektri
figure(i+3)
plot(taajuus, 10*log10(tehoavg))
title('Combined spectrum, every frame')
xlabel('Frequency (Hz)')
ylabel('PSD (dB/Hz)')

release(radio)
```