

Radiokaukokirjoituksen vastaanoton automatisoinnista

Ari Kosonen

Tampereen yliopisto
Tietojenkäsittelytieteiden laitos
Pro gradu -tutkielma
Kesäkuu 2002

Tampereen yliopisto

Tietojenkäsittelytieteiden laitos

Ari Kosonen: Radiokaukokirjoituksen vastaanoton automatisoinnista

Pro gradu-tutkielma, 59 sivua, 3 liitesivua

Kesäkuu 2002

Radiokaukokirjoitusta käytetään edelleen tiettyihin sovelluksiin radion lyhytaaltoalueilla, tietoliikennesatelliittiyhteyksien yleistymisestä huolimatta. Mekaanisista kaukokirjoittimista on kuitenkin siirrytty tietokoneen käyttämiseen kaukokirjoituksen vastaanotossa. Tässä tutkielmassa esitellään tavallisimpia radiokaukokirjoituksen muotoja sekä niiden vastaanottoon käytettäviä menetelmiä.

Tutkielman osana toteutettiin radiokaukokirjoituksen vastaanottojärjestelmä kahta erilaista lähestymistapaa käyttäen. Ensimmäinen toteutus perustuu ulkopuolisella laitteella suoritettavaan kaukokirjoitussignaalin demodulointiin, jossa vain kaukokirjoituskoodi puretaan tietokoneella. Jälkimmäinen menetelmä soveltaa digitaalista signaalinkäsittelyä siten, että vastaanotettava äänisignaali syötetään suoraan tietokoneen äänikortille ja signaalin demodulointikin suoritetaan ohjelmallisesti.

Digitaaliseen signaalinkäsittelyyn perustuva menetelmä osoittautui perinteistä lähestymistapaa huomattavasti joustavammaksi ja suorituskyvyltään vähintäänkin yhtä hyväksi. Koska erillistä demodulaattoria ei ole, vähenee myös vastaanottoon tarvittavien fyysisten laitteiden määrä. Lisäksi digitaalinen signaalinkäsittely mahdollistaa mm. vastaanottoparametrien automaattisen asettamisen.

Avainsanat ja -sanonnat: radiokaukokirjoitus, koodit, digitaalinen signaalinkäsittely.

Sisälllys

1.	Johdanto	1
2.	Katsaus lyhytaalloilla käytettäviin digitaalisen tiedonsiirron muotoihin	3
2.1.	Sähkötys	3
2.2.	Modulaatiomenetelmistä	5
2.2.1.	Vaihtotaajuusavainnus	6
2.2.2.	Vaiheensiirtoavainnus	7
2.3.	Perinteinen kaukokirjoitus Baudot-koodilla	8
2.4.	Virhesietoiset menetelmät	9
2.4.1.	SITOR ja AMTOR	10
2.4.2.	PACTOR	10
2.4.3.	G-TOR	12
2.5.	Vaihesiirtoon perustuvat menetelmät	13
2.5.1.	Clover	13
2.5.2.	PSK 31	14
2.6.	Muita menetelmiä	14
3.	Radiokaukokirjoituksen vastaanottojärjestelmä Baudot-koodille	16
3.1.	Radiokaukokirjoituksen vastaanoton peruseriaatteen	16
3.2.	Vaatimuksia radiokaukokirjoituksen vastaanottojärjestelmälle	17
4.	Ulkoista demodulaattoria käyttävä järjestelmä	18
4.1.	Ulkoinen AFSK-demodulaattori	18
4.2.	Demodulaattori modeemipiirillä toteutettuna	18
4.3.	Sarjaportin konfigurointi ja lukeminen	20
4.4.	Baudot-koodin dekoodaaminen	24
5.	Digitaaliseen signaalinkäsittelyyn perustuva järjestelmä	25
5.1.	DSP-pohjaisen järjestelmän rakenne	26
5.2.	Audiosignaalin lukeminen	27
5.3.	Suodattimien toteuttaminen ohjelmallisesti	28
5.3.1.	Suodattimien kertoimien määrittelystä	29
5.3.2.	Kaistanpäästösuodattimen kokeilua	33
5.4.	Signaalin demodulointi	34
5.5.	Dekoodaaminen	35

6.	Vastaanoton automatisoinnista	38
6.1.	Vastaanottimen ohjaaminen	38
6.1.1.	CAT-protokolla.....	38
6.1.2.	Lähetyslajin ja välitaajuussuodattimen asettaminen	40
6.1.3.	Taajuuden asettaminen	42
6.2.	Vastaanottoparametrien automaattinen asettaminen.....	42
6.2.1.	Fourier-muunnoksista	43
6.2.2.	Ikkunoinnista	46
6.2.3.	MARK- ja SPACE-signaalien hakeminen	48
6.3.	Kaukokirjoituslähetyksen tunnistaminen	49
6.3.1.	Lähetyksen tunnistaminen signaalin perusteella	49
6.3.2.	Lähetyksen tunnistaminen sisällön perusteella.....	50
7.	Toteutetusta järjestelmästä.....	51
7.1.	Menetelmien vertailua.....	51
7.2.	Järjestelmän arkkitehtuuri	51
7.3.	Käyttöliittymä.....	52
7.4.	Jatkokehitysmahdollisuuksia	53
8.	Yhteenveto	55

Viiteluettelo

Liitteet

1. Johdanto

Radiokaukokirjoitus on jokseenkin yhtä vanhaa kuin radioviestintä yleensäkin. Jo vuosikymmeniä käytössä olleet radiokaukokirjoituksen muodot ovat edelleen käytössä, mutta monia uusiakin muotoja on vuosien varrella kehitetty. Tietokoneiden käyttöönotto radiokaukokirjoituksessa on mahdollistanut mm. virheenkorjaavien menetelmien kehittämisen, mutta tietokone on korvannut mekaaniset kaukokirjoituslaitteet myös vanhempien menetelmien yhteydessä.

Digitaalista signaalinkäsittelyä on sovellettu radiokaukokirjoituksen vastaanotossa erityisesti 1990-luvulta lähtien. Sovellukset ovat kuitenkin keskittyneet pääasiassa audiosignaalin käsittelyyn ja demodulointiin. Digitaalinen signaalinkäsittely tarjoaa myös tehokkaita menetelmiä signaalin analysointiin. Tässä tutkielmassa selvitetään näiden menetelmien sopivuutta esimerkiksi vastaanottoparametrien automaattiseen asettamiseen radiokaukokirjoitussignaalille sopivaksi.

Tutkielmassa luodaan aluksi katsaus erilaisiin radiokaukokirjoituksen muotoihin. Luvussa 2 esitellään kaukokirjoituksessa käytettäviä modulaatiomenetelmiä sekä erilaisia keinoja, joilla on pyritty parantamaan radiokaukokirjoituksen häiriönsietokykyä ja tehokkuutta. Osa menetelmistä on jo jäänyt pois käytöstä, osa taas on pysynyt käytössä vuosikymmeniä suhteellisesta tehottomuudesta huolimatta.

Luvussa 3 tutustutaan radiokaukokirjoituksen vastaanoton peruseräisiin sekä asetetaan vaatimukset Baudot-kaukokirjoituksen vastaanottoon tarkoitetulle järjestelmälle. Seuraavaksi esitellään kahden erilaista lähestymistapaa käyttävän vastaanottojärjestelmän toteutus.

Luvussa 4 kuvataan järjestelmä, joka käyttää perinteiseen tapaan tietokoneen ulkopuolista laitetta kaukokirjoitussignaalin demodulointiin, siis äänisignaalin muuttamiseen digitaaliseen muotoon. Tässä menetelmässä datavirran lukeminen tietokoneen sarjaportilta toteutetaan ohjelmallisesti, samoin Baudot-koodin dekodaus.

Luvussa 5 esitellään järjestelmä, joka käyttää demodulointiin digitaalisen signaalinkäsittelyn keinoin toteutettuja suodattimia. Tässä järjestelmässä äänisignaali syötetään suoraan radiovastaanottimelta tietokoneen äänikortille, joten demoduloinnin ja dekodauksen kaikki vaiheet toteutetaan ohjelmallisesti. Samalla tutustutaan eräisiin digitaalisen signaalinkäsittelyn peruseräisiin ja menetelmiin.

Luvussa 6 kehitellään menetelmiä radiokaukokirjoituksen vastaanoton automatisointiin. Ensiksi esitellään toiminnot, joita tarvitaan radiovastaanottimen ohjaamisessa kaukokirjoitusjärjestelmästä käsin. Seuraavaksi perehdytään yh-

teen radiokaukokirjoituksen automatisoinnissa olennaiseen kohteeseen, eli vastaanottoparametrien automaattiseen asettamiseen. Radiokaukokirjoitus vaatii laitteilta stabiilisuutta. Lisäksi ongelmia on tuottanut vastaanottimen tai demodulaattorin ominaisuuksien asettaminen vastaanotettavalle signaalille sopiviksi. Digitaalinen signaalinkäsittely mahdollistaa näiden ongelmien ratkaisemisen. Yksi signaalinkäsittelyn perusmenetelmistä, diskreetti Fourier-muunnos ja erityisesti sen tehokas muoto, nopea Fourier-muunnos, mahdollistavat äänisignaalin taajuusspektrin analysoinnin. Analysoimalla vastaanottimesta kuuluvaa kaukokirjoitussignaalia voidaan vastaanottoparametrit asettaa automaattisesti. Lisäksi luvussa pohditaan mahdollisuuksia kaukokirjoitussignaalien tunnistamiseen ja hakemiseen automaattisesti, lähinnä digitaalista signaalinkäsittelyä soveltaen.

Luvussa 7 vertaillaan kahden toteutusvaihtoehdon ominaisuuksia ja esitellään toteutettavaksi valitun järjestelmän rakennetta. Lisäksi pohditaan mahdollisuuksia järjestelmän kehittämiseksi edelleen.

Luvussa 8 esitetään lyhyt yhteenveto tutkielman lopputuloksista.

2. Katsaus lyhytaalloilla käytettäviin digitaalisen tiedonsiirron muotoihin

Tiedonsiirron katsotaan olevan digitaalista, jos välitettävä signaali on luonteeltaan diskreettiä eli se saa vain ennalta määriteltyjä arvoja. Vastaavasti analoginen tiedonsiirto koostuu signaaleista, jotka saavat mielivaltaisia arvoja tietyllä vaihteluvälillä. [Stallings, 1994]

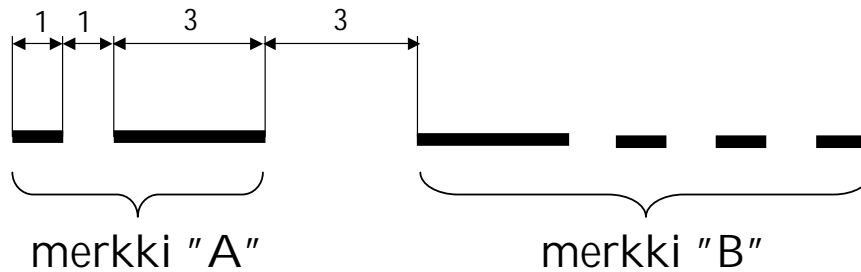
Radiokaukokirjoituksessa käytetään tiedon kuljettamiseen analogista radio-signaalia, jota joko katkotaan sovitulla tavalla tai moduloidaan äänisignaalilla. Menetelmien voidaan kuitenkin katsoa olevan digitaalisia, koska käytetty koodaus on luonteeltaan diskreettiä eli perustuu signaalin jonkin ominaisuuden, esimerkiksi amplitudin tai taajuuden, vaihteluun kahden ennalta määritellyn tason välillä.

Seuraavassa luodaan katsaus tavallisimpiin lyhytaalloilla käytettäviin digitaalisen tiedonsiirron muotoihin.

2.1. Sähkötys

Vanhin lyhytaalloilla käytetty digitaalinen tiedonsiirtomenetelmä on sähkötys. Sähkötöksessä sanoma koostuu merkeistä, joiden on sovittu vastaavan tiettyjä kirjaimia, numeroita tai muita merkkejä. Tunnetuin sähkötöksessä käytetty merkistö on Samuel Morsen 1800-luvulla kehittämä morsekoodi.

Morsekoodi koostuu pisteistä ja viivoista, joiden keskinäinen suhde on sovittu siten, että viiva on kolme kertaa pisteen pituinen. Vastaavasti on määriteltä myös pisteiden ja viivojen sekä erillisten merkkien väliset tauot. Saman merkin sisällä pisteet ja viivat erotetaan toisistaan tauolla, joka vastaa kestoltaan yhden pisteen kestoa. Kirjaimien väliin taas jätetään kolmen pisteen kestoa vastaava tauko. Lisäksi sanojen väliin jätetään tauko, joka on näitä pidempi, mutta tarkempi määrittely vaihtelee. Tyypillisesti sanojen välinen tauko on noin viiden pisteen kestoa vastaava [Morris, 1985]. Kuvassa 1 on esimerkki sähkötysmerkin elementtien suhteellisista pituuksista.



Kuva 1. Esimerkki morsekoodin merkkisuhteesta

Radiosähkötyksessä lähettimen lähettämää radiotaajuista kantoaaltoa katkotaan esimerkiksi sähkötysavaimella eli radiotaajuista energiaa päästetään antenniin avaimen kontaktien ollessa suljettuna. Vastaanottavassa päässä signaali muunnetaan ihmiskorvan kuultavissa olevalle äänitaajuudelle. Siirtonopeus on varsin vaatimaton, yleensä alle 2 merkkiä sekunnissa. Parhaimmatkin sähköttäjät pystyvät korvakuulolta vastaanottamaan ja tulkitsemaan vain muutamia merkkejä sekunnissa.

Vaikka sähkötystä on jo vuosikymmeniä lähetetty koneellisesti, se ei sovellu kovin hyvin koneelliseen vastaanottoon. Sähkötyssignaali on luonteeltaan diskreettiä, ts. se saa vain ennalta määrättyjä arvoja. Käytännössä kuitenkin merkit ovat keskenään eri mittaisia, eikä niiden tulkinta nopeusvaihteluista ja häiriöistä johtuen ole välttämättä yksiselitteistä. Sähkötysavaimella käsin lähetetyn sähkötyksen vastaanotto koneellisesti on vaikeaa, koska sähköttäjän "käsialasta" johtuen merkkien pituudet ja välit voivat suurestikin vaihdella. Koneellisesti lähetetyssä sähkötyksessä merkit ja välit ovat säännöllisen pituisia, mutta myös silloin pienikin häiriö riittää sekoittamaan vastaanoton.

Sähkötyksen koneellisen vastaanottoon on kehitetty menetelmiä, joista tunnetuin lienee *koherentti sähköty* (Coherent CW). Määritelmän mukaan koherentilla sähkötyksellä tarkoitetaan binäärisesti koodatun informaation, esimerkiksi morsekoodin lähettämistä siten, että vastaanottava pää tietää käytettävän taajuuden, pulssin pituuden sekä vaiheen ja käyttää näitä tietoja informaation tulkitsemisessa [Woodson, 1975]. Menetelmä kehitettiin alun perin parantamaan sähkötyssignaalin luettavuutta heikoilla signaalitasoilla. Myöhemmin havaittiin, että koherentti sähköty soveltuu säännönmukaisuutensa vuoksi hyvin myös koneellisesti vastaanotettavaksi.

Koherentissa sähkötyksessä käytetään tavallista morsekoodia, mutta merkkien pituus on vakioitu siten, että merkit koostuvat 0,1 sekunnin mittaisista yksiköistä. Piste on yhden yksikön, siis 0,1 sekunnin, mittainen pulssi ja pisteiden välissä oleva tauko on myös saman mittainen. Viiva vastaavasti on kolmen yksikön mittainen eli merkkisuhde vastaa ideaalista sähkötyksen merkkisuhdetta.

Nopeus vastaa tavallisen sähkötyksen 120 merkin minuuttinopeutta ja on siten vielä helposti ihmiskorvinkin vastaanotettavissa.

Koherentin sähkötyksen koneellisessa vastaanotossa vastaanottava laite synkronoidaan lähettävään signaaliin siten, että se "tietää" milloin merkin pitäisi alkaa. Käyttäen taajuusvakaita laitteita sekä lähettävässä että vastaanottavassa päässä, voidaan käyttää kaistanleveydeltään huomattavasti normaalia sähkötyssignaalia kapeampaa lähetettä. Näin lähetteen signaali-kohinasuhde saadaan paremmaksi. Alun perin signaalit synkronoitiin siten, että kapeakaistaisesta suodattimesta saadulla signaalilla ohjattiin vastaanottimessa äänitaajuusoskillaattoria, jolla luotiin sähkötykset uudelleen ihmiskorvin vastaanotettaviksi.

Kun 1970-luvulla tehtiin ensimmäisiä kokeita koherentilla sähkötyksellä, oli riittävän taajuusvakaiden laitteiden toteuttaminen kallista ja menetelmä ei yleistynyt. Viime vuosina radiolaitteiden kehittymisen ja erityisesti digitaaliseen signaalinkäsittelyyn soveltuvien laitteiden yleistymisen myötä koherentin sähkötyksen perusidea on tullut jälleen ajankohtaiseksi. Menetelmää on käytetty heikkojen signaalien vastaanottoon tietyissä erikoissovelluksissa, esimerkiksi kuun kautta heijastamalla pidettävissä yhteyksissä VHF-taajuuksilla. Morsekoodin ohella näissä kuitenkin käytetään myös tehokkaampia ja yksiselitteisempiä koodauksia.

2.2. Modulaatiomenetelmistä

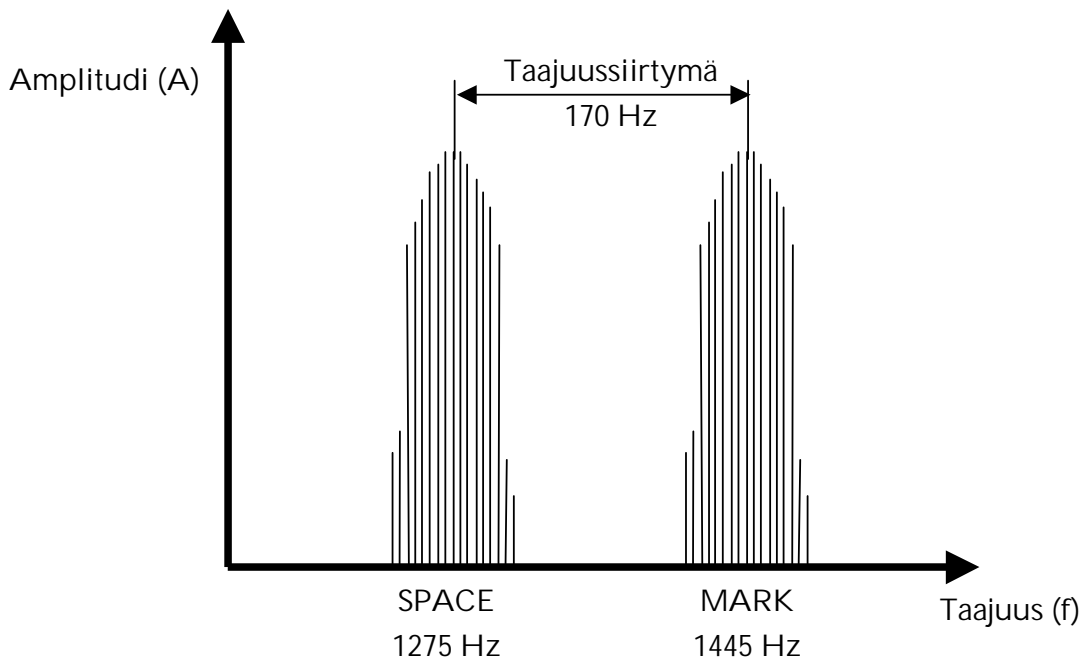
Moduloinnissa on kyse *kantoaallon* (carrier wave) muokkaamisesta siirrettävää informaatiota edustavalla *moduloivalla signaalilla* (modulating signal). Tuloksena syntyvä moduloitu signaali kuljettaa siirrettävän informaation. Vastaanotavassa päässä signaali demoduloidaan ja siten saadaan siirretty informaatio jälleen esille. [Carlson *et al.*, 2002]

Sähkötyksessä radiotaajuista kantaaltoa ei moduloida, vaan sitä katkotaan joko sähkötysavaimella tai koneellisesti ja tällä tavoin saadaan morsemerkit muodostettua. Sähkötyssignaali on varsin kapeakaistainen, vain muutamia satoja hertsejä. Kapeallekin taajuuskaistalle mahtuu siis useita sähkötyssignaaleja, joiden taajuus poikkeaa vain hieman toisistaan. Ihmisen kuuloaistin erinomaisesta erottelukyvystä johtuen sähkötyksen vastaanotto on kuitenkin mahdollista häiriöisissäkin olosuhteissa. Koneellisesti vastaanotettuna sähkötyksen häiriönsietokyky on kuitenkin varsin heikko, koska informaation siirrossa käytettävät signaalit voivat muuttua tai hävitä. Lisäksi sähkötyksen tiedonsiirtonopeus on varsin vaatimaton. Siirtonopeuden kasvattamiseksi ja häiriönsietokyvyn parantamiseksi on kehitelty erilaisia modulaatiomenetelmiä.

2.2.1. Vaihtotaajuusavainnus

Radiokaukokirjoituksessa on modulaatiomenetelmänä käytetty pääasiassa vaihtotaajuusavainnusta, joka voi olla joko *äänitaajuusavainnusta* (AFSK, Audio Frequency Shift Keying) tai *radiotaajuusavainnusta* (FSK, Frequency Shift Keying). Äänitaajuusavainnetussa radiokaukokirjoituksessa käytetään puhelii-kenteeseen tarkoitettua radiolähetintä, jonka mikrofonilinjaan syötetään MARK- ja SPACE-merkkejä vastaavia äänitaajuuksia. Radiotaajuusavainnuk-sessa puolestaan vaihdellaan käytettävien kantaaltojen radiotaajuutta.

Vaihtotaajuusavainnetussa radiokaukokirjoituksessa merkit koostuvat nol-lista ja ykkösistä, jotka esitetään äänitaajuisina signaaleina. Näitä kutsutaan SPACE-signaaliksi (nolla) ja MARK-signaaliksi (ykkönen). Signaalit ovat kes-kenään lomitettuja, samanaikaisesti siis lähetetään vain toista signaaleista. Pu-heliikenteeseen käytettävän *yksisivunauhamoduloidun* (SSB, Single Sideband Modulation) lähettimen äänikaistan leveys on yleensä alle 3 kilohertsiä, joten äänitaajuusavainnuksessa käytettävät äänitaajuusparit on valittu siten, että ne molemmat mahtuvat tähän kaistaan. Kuvassa 2 on esimerkki tyypillisestä ly-hytaalloilla käytettävästä MARK- ja SPACE-taajuusparista.



Kuva 2. MARK- ja SPACE-signaalit ja niiden välinen taajuussiirtymä.

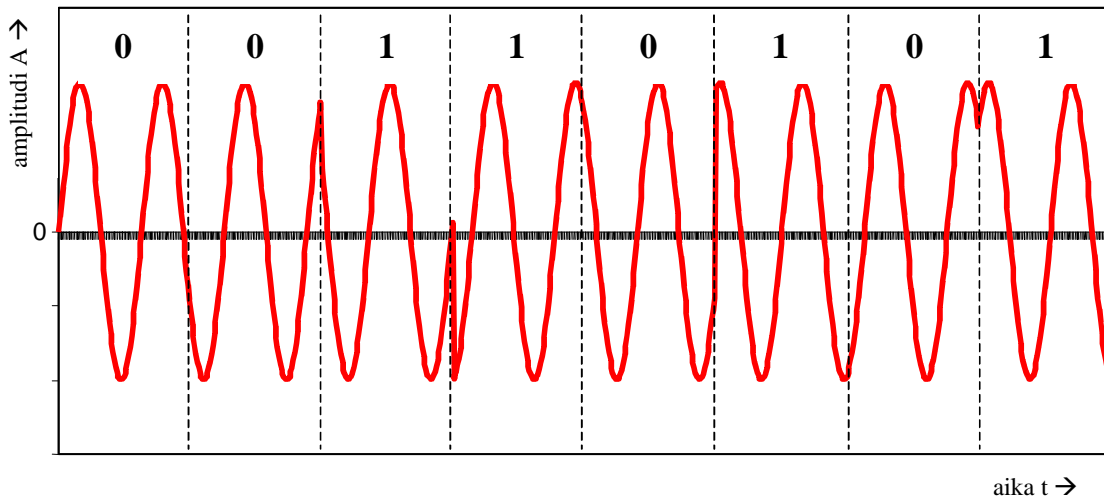
Vaihtotaajuusavainnuksen symbolinopeus baudeina määritellään yhden pulssin keston käänteislukuna. Lyhytaaltotaajuuksilla käytettävät kaistanle-veydet rajoittavat käyttökelpoisen nopeuden alle 300 baudin, kun käytetään kaksitaajuisia vaihtotaajuusavainnusta. Myös useampia äänitaajuuksia käyttä-

viä vaihtotaajuusmodulaatiomenetelmiä (MFSK, Multitone Frequency Shift Keying) on käytössä.

2.2.2. Vaiheensiirtoavainnus

Vaihtotaajuusavainnuksen ohella toinen yleisesti käytetty modulaatiomenetelmä on *vaiheensiirtoavainnus* (PSK, Phase Shift Keying). Vaiheensiirtoavainnuksessa lähetettävän signaalin vaihetta käytetään informaation siirtoon. Yleensä kyseessä on vaiheen muutokseen perustuva avainnus. Tällöin bittiä 0 vastaa signaalipurske, joka on samassa vaiheessa kuin edellinen lähetetty purske ja bittiä 1 vastaa purske, joka on vastakkaisessa vaiheessa edelliseen purskeeseen nähden. Menetelmästä käytetään myös nimitystä *vaihe-eroavainnus* (DPSK, Differential Phase-Shift Keying). Mikäli modulointiin käytetään kahta toisilleen vastakkaista vaihetta, kyseessä on *binäärinen vaiheensiirtoavainnus* (BPSK, Binary Phase-Shift Keying). Kun puhutaan *nelitasoisesta vaiheensiirtoavainnuksesta* (QPSK, Quadrature Phase-Shift Keying), käytetään neljää vaihetta, jotka poikkeavat toisistaan 90 astetta [Stallings, 1994].

Kuvassa 3 on esitetty bittijonon 00110101 modulointi binäärisellä PSK:lla. Mikäli lähetettävä bitti on nolla, se lähetetään samassa vaiheessa kuin edellisenkin bitti. Jos lähetettävä bitti on ykkönen, vaihetta käännetään 180 astetta. Näin ollen vaihe muuttuu myös peräkkäisten ykkösbittien välillä, mutta se ei muutu, mikäli ykköstä seuraakin nolla.



Kuva 3. Esimerkki moduloinnista PSK:lla

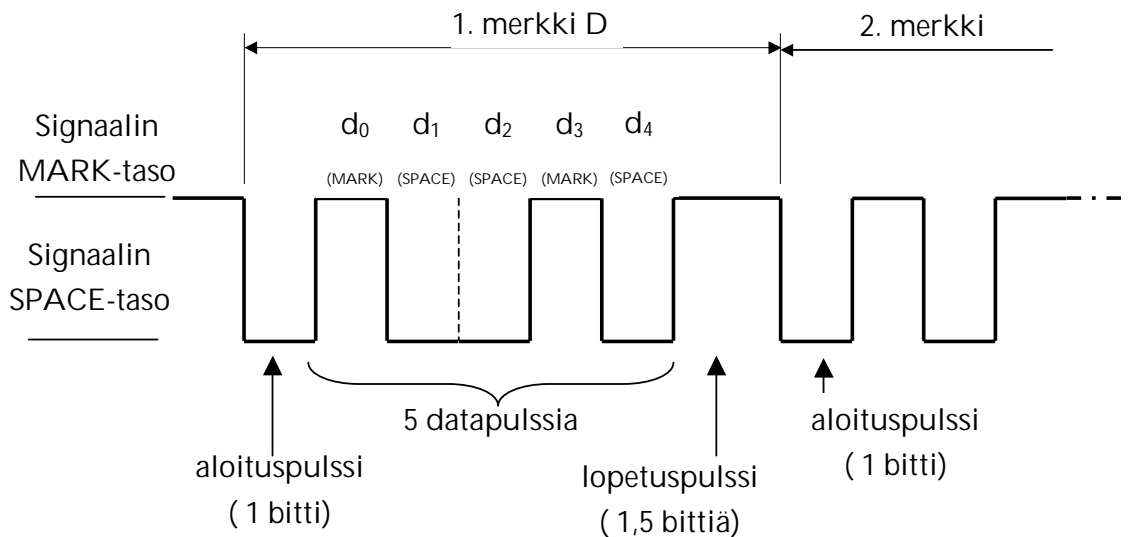
Viime vuosina myös vaiheensiirtoavainnusta on sovellettu uusissa radio-kaukokirjoitusmenetelmissä, kuten esimerkiksi Cloverissa (ks. kohta 2.5.1) ja PSK31:ssä (ks. kohta 2.5.2). Erityisesti PSK31 on yleistynyt ja jossakin määrin

korvannut perinteisen Baudot-koodatun radiokaukokirjoituksen varsinkin sovelluksissa, joissa ei vaadita virheenkorjausta [Ford, 1999].

2.3. Perinteinen kaukokirjoitus Baudot-koodilla

Kaukokirjoitus, kuten sähkötyöskin, on alun perin kehitetty käytettäväksi lankalennätinverkoissa, mutta sittemmin otettu käyttöön myös radioteitse tapahtuvassa viestinvälityksessä. Sähkötyksestä poiketen kaukokirjoitus on tarkoitettu vain koneella vastaanotettavaksi. Tästä johtuen kaukokirjoituksessa käytetty merkistö on deterministisempi, koska kukin merkki on saman mittainen ja saman muotoinen.

Tunnetuin ja pisimpään käytössä ollut kaukokirjoituskoodi on Baudot-koodi, joka tunnetaan myös nimellä "International Telegraph Alphabet No. 2." Baudot-koodi on viisibittinen, kunkin merkin alussa on yksi aloitusbitti ja lopussa 1,5 lopetusbittiä. Käytettäessä 50 baudin nopeutta on yhden elementin, siis pulssin, pituus 20 millisekuntia [Renton, 1974]. Kuvassa 4 on esimerkki kirjaimen D muodostamisesta Baudot-koodilla.



Kuva 4. Baudot-koodattu kaukokirjoitussignaali kirjaimelle D

Koska varsinaisen datan esittämiseen on käytettävissä vain viisi bittiä, Baudot-koodilla voidaan esittää vain 32 eri merkkiä. Jotta sekä kirjaimien että numeroiden ja välimerkkien esittäminen olisi mahdollista, on kirjaimet ja muut merkit jaettu kahdelle eri tasolle. Sama merkki voi siis tarkoittaa sekä kirjainta että numeroa. Tasonvaihto osoitetaan erityisillä tasonvaihtomerkeillä: FIGS-merkki kertoo että siirrytään kirjaintasolta numerotasolle, jolla merkit ovat numeroita tai välimerkkejä. LTRS-merkillä taas palataan takaisin kirjaintasolle.

Kuriositeettina mainittakoon, että Venäjällä on käytetty kolmitasoista merkistöä, jossa kyrilliset kirjaimet on esitetty omalla tasollaan koodin muuten vastatessa tavallista Baudot-koodia. Myös muita variaatioita esiintyy. Normaalin Baudot-koodin merkistö on esitetty liitteessä 1.

Baudot-koodin hyötysuhde on melko huono, koska kukin 7,5-bittinen merkki kuljettaa vain 5 bittiä varsinaista dataa. Lisäksi tasonvaihtomerkit ovat puhtaasti kontrollidataa ja niiden hukkuminen aiheuttaa dekodauksen jäämisen väärälle tasolle seuraavaan tasonvaihtomerkkiin saakka. Tasonvaihtojen katoamisen aiheuttamia virheitä voidaan vähentää lähettämällä tasonvaihtomerkki ennen jokaista siirrettävää merkkiä tai jokaisen sanan jälkeen. Tämä tietysti heikentää hyötysuhdetta entisestään.

Myös muita koodeja, kuten tavallista 7-bittistä ASCII-koodia, on käytetty kaukokirjoitukseen. Baudot-koodi on kuitenkin edelleen tavallisin lyhytaalloilla käytetyistä perinteisistä ei-virhesietoisista radiokaukokirjoituskoodeista. Käytössä on kuitenkin useita eri nopeuksia, joista tavallisimmat ovat 50, 75, 100 ja 300 baudia. Myös erilaisia MARK- ja SPACE-signaalien välisiä taajuussiirtymiä esiintyy, mm. 170, 450 ja 850 hertsiä. Esimerkiksi säätiedotuksia laivaliikenteelle lähetävä saksalainen kaukokirjoitusasema DDK9 (Deutscher Wetterdienst Hamburg) taajuudella 10102,8 kHz käyttää 50 baudin Baudot-kaukokirjoitusta 450 hertsin erotuksella [DWD, 2002]. Radioamatöörit taas käyttävät yleisimmin 45,45 baudin nopeutta 170 hertsin erotuksella [Ford, 1999]. Kansainvälisessä Telex-verkossa on käytetty 50 baudin nopeutta, joka on ollut tavallisin myös radiokaukokirjoituspalveluissa [Renton, 1974].

2.4. Virhesietoiset menetelmät

Perinteinen kaukokirjoitus Baudot-koodia käyttäen ei ole virhesietoista. Varsinkin pitkillä yhteysväleillä muut radiolähetteen ja säätimet aiheuttavat häiriöitä. Myös ilmakehän kerrosten ominaisuuksista johtuvat muutokset radioaaltojen etenemisessä aiheuttavat signaalin häipymistä.

Koska monissa viestintäsovelluksissa on välttämätöntä, että viesti saadaan virheettömästi perille, on kehitetty erilaisia virhesietoisia menetelmiä. Useimmat näistä vaativat kuittauksen vastaanottavasta päästä, joten myös vastaanottaja tarvitsee radiolähtetimen. Kuittaukseen perustuvaa virheenkorjausta kutsutaan *ARQ-moodiksi* (Automatic Repeat Request). Monista virhesietoista menetelmistä on olemassa myös *FEC-moodia* (Forward Error Correction) käytäviä versioita. FEC-moodia käytetään useammalle vastaanottajalle samanaikaisesti tarkoitettuihin lähetyksiin ja lähetyksiin joita ei pystytä kuittaamaan. FEC-moodia käytettäessä kukin purske sisältää koodauksen, jota käyttäen pystytään korjaamaan yksittäisiä puuttuvia bittejä vielä vastaanottavassa päässä. Erityisen häiriöisissä olosuhteissa kukin purske lähetetään mahdollisesti useampaan

kertaan. Tarkastussumma perustuu yleensä CRC-menetelmään (Cyclic Redundancy Check).

Monet uusista virhesietoisista radiokaukokirjoitusmenetelmistä ovat laitevalmistajien kehittämiä ja siten sidoksissa kyseisen valmistajan laitteistoihin. Tällaisia menetelmiä ovat esimerkiksi PACTOR ja G-TOR.

2.4.1. SITOR ja AMTOR

Vanhimpia virhesietoisista radiokaukokirjoitusmenetelmistä ovat lähinnä laivaliikenteen käyttämä SITOR (Simplex Teleprinting Over Radio) ja radioamatöörien AMTOR (Amateur Teleprinting Over Radio), jotka on otettu käyttöön 1970-luvun lopulla. Molemmat näistä perustuvat CCIR:n suositukseen 476-3 [Wade, 1987]. Näissä menetelmissä data lähetetään vakiomittaisina purskeina, jotka sisältävät hyötykuorman lisäksi myös tarkastussumman. Mikäli vastaanottavassa päässä todetaan, että tarkastussumma ei täsmää, pyydetään purskeen uudelleenlähetystä NACK-sanomalla (Not Acknowledged). Oikein vastaanotetut sanomat kuitataan ACK-sanomalla (Acknowledged). Puuttuva tai korruptoitunut ACK-sanoma oletetaan uudelleenlähetyspyynnöksi. Näiden menetelmien hyötysuhde ei ole optimaalinen, koska koko purske lähetetään aina uudelleen, vaikka vain yksi bitti olisi virheellinen.

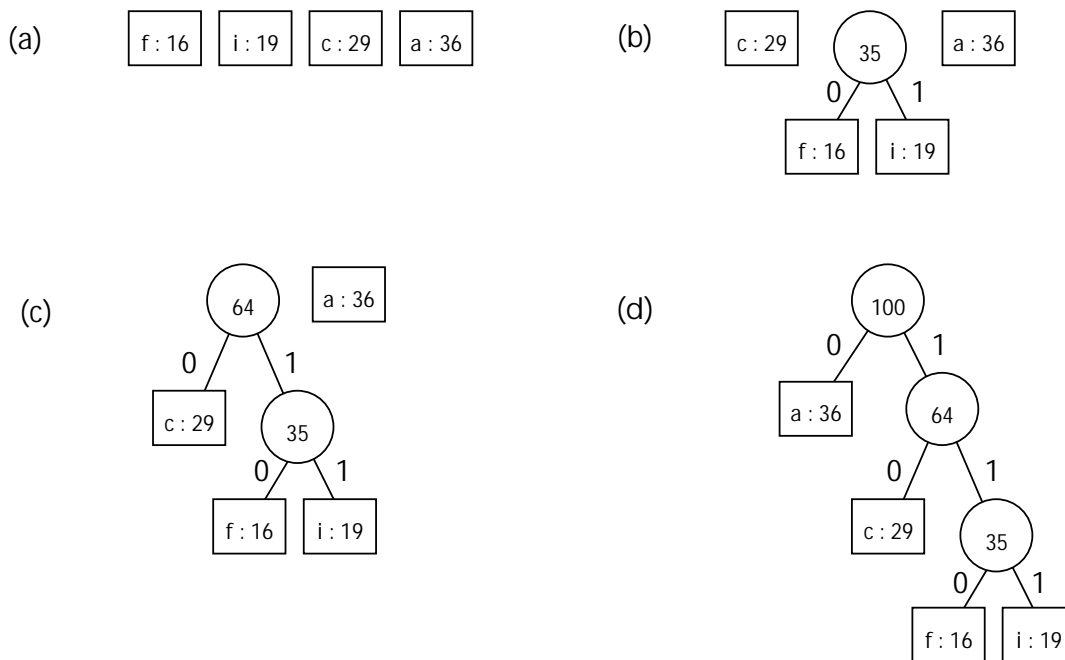
SITOR ja AMTOR ovat jossakin määrin edelleen käytössä, mutta uusien pakettimuotoisten menetelmien kehittymien 1990-luvulla on syrjäyttänyt varsinkin AMTOR:n lähes kokonaan. AMTOR-lähetteellä käytetään yleensä ASCII-koodia ja 100 baudin lähetysnopeutta. Näistä lähetyslajeista käytetään kuittaavan moodin (AMTOR ARQ tai SITOR-A) ohella myöskin kuittaamatonta moodia (AMTOR FEC, SITOR-B tai NAVTEX).

2.4.2. PACTOR

Saksalaiset radioamatöörit Hans-Peter Helfert ja Ulrich Strate kehittivät 1990-luvun alussa PACTOR:n (Packet Teleprinting Over Radio), joka muistuttaa edellä esiteltyä AMTOR:a [Helfert and Strate, 1991]. Jos PACTOR:a käytettäessä vastaanotetusta datalohkosta osa todetaan virheettömäksi, se tallennetaan muistiin ja kuittaussanomassa pyydetään vain virheettiset osat uudelleen lähetettäväksi. Tätä toistetaan kunnes koko datalohko on saatu koottua virheettömästi. Älykkäämmän virhekorjauksen lisäksi PACTOR osaa säätää käytettävää siirtonopeutta häiriötilanteen mukaan. Hyvissä olosuhteissa nopeus voi nousta 200 baudiin. Välitysnopeutta on edelleen nostettu käyttämällä ns. Huffman-koodausta, jossa harvemmin käytetyt merkit esitetään useammalla bitillä kuin yleisemmät merkit.

Huffman-koodaus on varsin yleisesti käytetty ja tehokas pakkausmenetelmä, jossa koodisanan pituus vaihtelee kunkin merkin yleisyyden mukaan siten, että usein esiintyvälle merkille käytetään lyhyempää koodisanaa kuin harvemmin esiintyvälle merkille [Cormen *et al.*, 1997]. Huffman-koodi muodostetaan algoritmilla, jossa merkkistö järjestetään esiintymistodennäköisyytensä mukaan puurakenteeseen siten, että kaksi pienimmän painoarvon omaavaa merkkiä yhdistetään alipuuksi. Alipuun juuren painoarvo on kyseisen puun lehtien, siis merkkien esiintymistodennäköisyyksien, summa. Puu järjestetään siten, että vasen lehti on painoarvoltaan oikeaa pienempi ja sinne johtava kaari merkitään nolllalla. Oikeaan lehteen johtava kaari merkitään vastaavasti ykkösellä. Tästä edetään yhdistelemällä aina kaksi painoarvoltaan pienintä solmua, kunnes koko merkkistö on saatu järjestettyä puurakenteeseen. Koodi saadaan kuhunkin merkkiin johtavan polun kaarien tunnuksista.

Kuvassa 5 on esitetty Huffman-koodin muodostaminen nelimerkkiselle merkkistölle vaiheiden (a)...(d) kautta.



Kuva 5. Huffman-koodin muodostaminen nelimerkkiselle merkkistölle

PACTOR:ssa käytetty Huffman-koodaus perustuu kirjainten yleisyyteen saksan kielessä. Esimerkiksi suhteellisen yleinen e-kirjain esitetään kolmella bitillä (011), kun taas harvinaisemman X-kirjaimen esittämiseen käytetään peräti 13 bittiä (0001010111011). Kun ASCII-koodissa saadaan kukin englannin kielen merkki esitettyä 8 bitillä, on PACTOR:n Huffman-koodattujen merkkien keskimääräinen pituus alle 5 bittiä [Ford, 1999].

PACTOR:sta on myöhemmin kehitetty lähinnä sotilaallisiin ja kaupallisiin tarkoituksiin parannettu versio, PACTOR II. Tässä suurinta siirtonopeutta on saatu kasvatettua kuusinkertaiseksi alkuperäiseen PACTOR:iin nähden ja Huffman-koodauksen lisäksi käytetään ns. *pseudo-Markov-pakkausta* (PMC, Pseudo-Markov Compression).

Pseudo-Markov-pakkaus perustuu Markovin ketjuihin, jotka ovat mm. Huffman-koodauksen ohella eräitä keskeisimpiä tiedon pakkaukseen käytettyjä menetelmiä [Williams, 1991]. Markov-koodauksessa kullekin kirjaimelle on määritelty todennäköisin seuraava kirjain. Esimerkiksi w-kirjaimen esiintymistodennäköisyys e:n jälkeen ei ole yhtä suuri, kuin vaikkapa i-kirjaimen. Koska Markov-koodauksen vaatima vertailutaulu kullekin kirjaimelle olisi varsin suuri, on PACTOR II:een valittu käytettäväksi pseudo-Markov-koodaus, jossa taulukon kokoa on rajoitettu.

Alkuperäisestä PACTOR:sta poiketen PACTOR II ei käytä pelkkää vaihtotaajuusavainnusta, vaan häiriöolosuhteista riippuen joko vaihtotaajuusavainnusta tai erilaisia vaihesiirtoon perustuvia modulaatiomenetelmiä [Rink and Helfert, 1995].

PACTOR III on uusin tämän perheen versioista. Se on PACTOR II:een verrattuna tyypillisesti noin 3,5 kertaa nopeampi ja myös jonkin verran häiriösitoisempi. Tavallisen SSB-yhteyden 2,4 kHz:n kaistaleveydellä suurin mahdollinen nopeus on noin 5200 bittiä sekunnissa protokollan sisältämää pakkausmenetelmää käyttäen. Fyysinen maksiminopeus on 3600 bittiä sekunnissa. PACTOR III on yhteensopiva edellisten PACTOR-versioiden kanssa siten, että yhteyden muodostuksessa protokolla neuvottelee PACTOR-version, jota molemmat vasta-asetat pystyvät käyttämään [SCS, 2002].

2.4.3. G-TOR

Toinen 1990-luvulla esitelty virhesietoinen kaukokirjoitusmenetelmä on G-TOR (Golay-coded Teleprinting Over Radio), joka perustuu amerikkalaiseen sotilasstandardiin MIL-STD-188-141A. G-TOR on synkronoitu menetelmä, jossa data on pakattu 1,92 sekunnin mittaisiin kehyksiin, joiden välissä on 0,48 sekunnin mittainen tauko. Kehykset ovat aina kestoaltaan vakiomittaisia, mutta riippuen käytettävästä siirtonopeudesta yksi kehys voi sisältää joko 192, 384 tai 576 bittiä, vastaten nopeuksia 100, 200 tai 300 merkkiä sekunnissa [Ford, 1999]. Käytettävää nopeutta vaihdetaan häiriöolosuhteiden mukaan automaattisesti. Siirrettävän datan lisäksi kehyksessä kuljetetaan tarkastussumma sekä statuskenttä. Statuskenttä sisältää tiedon siirrettävän datan formaatista (8-bittinen ASCII tai Huffman-koodattu), kehyksen numeron sekä muuta ohjaustietoa. G-TOR:n käyttämä Huffman-koodaus poikkeaa PACTOR:n käyttämästä siten, että merkkien pituuksia on painotettu englannin kielen mukaan. G-TOR:n vir-

heenkorjaus perustuu Golay (23,12) -koodaukseen, joka kykenee korjaamaan enintään kolme virhettä 23-bittisessä koodisanassa [Peterson and Weldon, 1981].

G-TOR ei ole juurikaan yleistynyt. Syynä lienee se, että G-TOR on ollut sidoksissa yhden laitevalmistajan toteutukseen ja toisaalta sen suorituskyky häviää esimerkiksi PACTOR:n uudemmille versioille.

2.5. Vaihesiirtoon perustuvat menetelmät

Kaikki edellä esitellyt menetelmät käyttävät vaihtotaajuusavainnusta modulaatiomenetelmänään. Erityisesti 1990-luvun alkupuolelta lähtien on kokeiltu myös erilaisia vaihesiirtoon perustuvia tekniikoita. Seuraavassa esitellään näistä menetelmistä muutamia, jotka ovat tulleet käyttöön lähinnä digitaalisen signaalinkäsittelyn myötä.

2.5.1. Clover

Amerikkalainen Ray Petit kehitti 1990-luvun alussa Clover-menetelmän. Tavoitteena oli luoda tehokas menetelmä erityisesti binääritiedostojen siirtoon lyhytaaltoradioyhteyksiä käyttäen. Cloverin modulaatiomuoto on varsin monimutkainen. Se koostuu neljästä 125 hertsia leveästä äänipulssista, jotka ovat 125 hertsin etäisyydellä toisistaan. Vain yhtä äänitaajuutta lähetetään kerrallaan ja kukin pulssi on kestoaltaan 8 millisekuntia. Yksi datakehys sisältää neljä äänipulssia, joten kehyksen kesto on 32 millisekuntia ja modulaationopeudeksi saadaan 31,25 symbolia sekunnissa. Varsinainen datan siirto tapahtuu vaihtelemalla pulssien vaihetta tai amplitudia. Clover koostuu itse asiassa kuudesta erilaisesta modulaatiomuodosta, joista yksinkertaisimmat perustuvat pelkkään vaiheensiirtoavainnukseen, monimutkaisimmat taas jopa 16-tasoisien vaiheensiirtoavainnuksen ja 4-tasoisien amplitudiavainnuksen yhdistelmään. Clover valitsee sopivimman modulaatiomenetelmän riippuen radioyhteyden laadusta. Tehokkaimmalla modulaatiomenetelmällä päästään 750 bitin sekuntinopeuteen, kun taas häiriöisissä olosuhteissa käytettävällä pelkällä PSK-modulaatiolla päästään 62,5 - 375 bittiin sekunnissa riippuen siitä, kuinka monta eri äänisignaalin vaihetta pystytään käyttämään.

Cloverin kuittaamattomassa FEC-muodossa virheenkorjaus perustuu Reedin-Solomonin koodaukseen, joka soveltuu hyvin purskeisten virheiden korjaamiseen monitaajuisissa modulaatiomenetelmissä [Carlson *et al.*, 2002]. Cloverissa on mahdollista käyttää myöskin kuittaavaa ARQ-muotoa. Tällöin lähettävä osapuoli saa kuitaussanomassa tietoa myöskin häiriösuhteesta sekä vaihevirheistä vastaanottavassa päässä, ja tarvittaessa pystytään vaihtamaan käytettävä modulaatiomenetelmä olosuhteisiin paremmin sopivaksi.

Cloverista on kehitetty uusi, noin neljä kertaa entistä nopeampi versio, Clover-2000. Se käyttää hieman leveämpää taajuuskaistaa, ja yksi datakehys koostuu kahdeksasta 2 millisekunnin kestoisesta äänipulssista. Tällöin modulaationopeudeksi muodostuu 62,5 symbolia sekunnissa [ARRL, 2000]. Tehokkuudestaan ja häiriönsietoisuudesta huolimatta Cloverin eri versiot eivät ole juurikaan yleistyneet.

2.5.2. PSK31

Englantilainen Peter Martinez ryhtyi 1990-luvun lopulla kehittämään uutta menetelmää korvaamaan jo vuosikymmeniä käytetyn Baudot-koodatun radiokaukokirjoituksen. Tavoitteena oli saada aikaiseksi kapeakaistainen ja häiriösetoinen menetelmä. Tuloksena syntyi PSK31, joka on vaiheensiirtomodulaatiota käyttävä menetelmä, jonka symbolinopeus on 31,25 baudia. PSK31 on erittäin kapeakaistainen, vain 80 hertsiä. Modulaatiomenetelmä on joko binäärinen vaiheensiirtoavainnus (BPSK) ilman virheenkorjausta tai nelitasoinen vaiheensiirtoavainnus (QPSK) virheenkorjauksella. Virheenkorjaukseen käytetään konvoluutiokoodausta ja Viterbi-dekoodausta.

PSK31:n käyttämä koodi on nimeltään Varicode, jossa englannin kielessä tavallisimmat merkit on esitetty lyhyemmillä ja harvinaisemmat pitemmillä koodeilla. Koodi on muodostettu siten, että minkään merkin koodi ei sisällä kahta peräkkäistä nollaa. Sen sijaan kahden peräkkäisen merkin väliin sijoitetaan erotuskoodi, joka koostuu vähintään kahdesta nollasta. Varsinaiset merkit alkavat aina ykkösellä ja ne myös aina päättyvät ykköseen.

Tällä hetkellä näyttää siltä, että varsinkin radioamatööripiireissä PSK31 on vahvasti yleistymässä ja saattaa osittain korvata perinteisen Baudot-koodatun radiokaukokirjoituksen [ARRL, 2000].

2.6. Muita menetelmiä

Edellä kuvattujen lisäksi myös monia muita menetelmiä on kokeiltu. Esimerkkinä mainittakoon puolalaisen Pawel Jaločan kehittämä MT-63, joka on peräti 64 eri äänitaajuutta käyttävä erittäin virhesietoinen vaihtotaajuusavainnusmenetelmä. Se on kuitenkin varsin leveäkaistainen, eikä siinä mielessä kovin hyvin sovi ylikuormitetuille lyhytaaltotaajuuksille [Ford, 1999]. Myös erilaisia vaihtotaajuusavainnuksen ja vaiheensiirtoavainnuksen yhdistelmiä on esitelty, mutta toistaiseksi mikään niistä ei ole saavuttanut mainittavaa suosiota, joitakin erikoissovelluksia lukuun ottamatta.

Yksi uusimpia tulokkaita on FSK441, neljää äänitaajuutta käyttävä vaihtotaajuusavainnusmenetelmä, jonka nopeus on 441 baudia. Tätä käytetään VHF-taajuuksilla meteorisironnan ilmakehään synnyttämien ionisoitujen vanojen

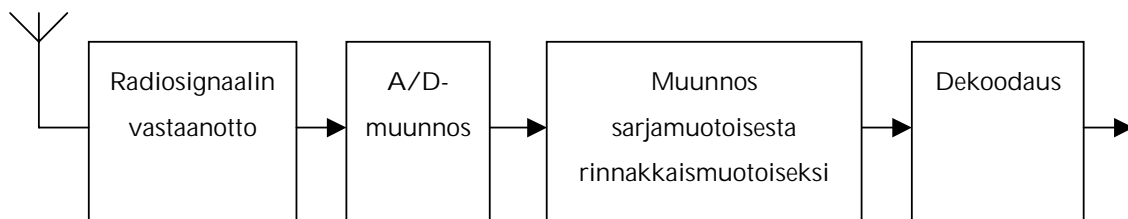
kautta tapahtuvaan yhteydenpitoon. Tähän tarkoitukseen eivät vaiheensiirtoon perustuvat menetelmät sovellu, johtuen Doppler-siirtymän ja heijastusetäisyyden nopean vaihtelun aiheuttamista vaihepoikkeamista. [Taylor, 2001]

3. Radiokaukokirjoituksen vastaanottojärjestelmä Baudot-koodille

Tässä luvussa esitellään radiokaukokirjoituksen vastaanoton peruseräatteen sekä asetetaan vaatimuksia radiokaukokirjoituksen vastaanottojärjestelmälle, joka toteutetaan osana tätä tutkielmaa.

3.1. Radiokaukokirjoituksen vastaanoton peruseräatteen

Kohdissa 2.3 ja 2.4 esiteltyt vaihtotaajuusavainnukseen perustuvat menetelmät poikkeavat toisistaan lähinnä käytetyn koodauksen ja mahdollisen yhteysprotokollan sekä virheenkorjausominaisuuksien osalta. Kun tarkastellaan lähempänä radioyhteyttä olevia toimintoja, ovat samat vaiheet erotettavissa radiokaukokirjoituksen vastaanotossa menetelmästä tai koodauksesta riippumatta. Vaiheet on esitetty kuvassa 6.



Kuva 6. Radiokaukokirjoituksen vastaanoton vaiheet

Vastaanotossa ensimmäinen vaihe on radiotaajuisen signaalin muuntaminen äänisignaaliksi. Tämä tapahtuu radiovastaanottimessa. Seuraavaksi analoginen äänisignaali saatetaan digitaaliseen muotoon. Kyse on siis A/D-muunnoksesta. Tämä voidaan toteuttaa joko erillisellä demodulaattorilla tai esimerkiksi tietokoneen äänikortilla.

A/D-muunnoksen jälkeen tuloksena on sarjamuotoinen bittivirta, josta täytyy erottaa yksittäistä merkkiä vastaavat rinnakkaismuotoiset tavut.

Viimeisenä vaiheena yksittäiset tavut dekodataan merkeiksi käytetyn koodin mukaisesti ja esitetään tekstimuodossa. Lopputulos voidaan esittää tietokoneen näytöllä taikka ohjata kirjoittimelle tai tiedostoon.

3.2. Vaatimuksia toteutettavalle kaukokirjoituksen vastaanottojärjestelmälle

Osana tätä tutkielmaa toteutettava radiokaukokirjoituksen vastaanottojärjestelmä on tarkoitettu Baudot-koodia käyttävän radiokaukokirjoituksen vastaanottoon. Baudot-koodi on 5-bittistä, lopetusbittejä on 1,5 eikä pariteettitarkastusta käytetä. Järjestelmän tulee pystyä dekodamaan Baudot-koodia ainakin 45,45 sekä 50 baudin nopeuksilla. Lisäksi järjestelmän täytyy pystyä vastaanottamaan ainakin signaaleja, joissa MARK- ja SPACE-tasojen välinen taajuussiirtymä on 150 tai 450 hertsiä. Nopeus ja taajuussiirtymä voidaan asettaa joko ohjelmallisesti tai demodulaattoria säätämällä. Dekoodattu teksti esitetään tietokoneen tai erillisen päätteen näytöllä. Käyttöliittymä on merkkipohjainen ja sitä on pystyttävä käyttämään VT-100-yhteensopivalla päätteellä tai pääteohjelmalla.

Järjestelmä voi käyttää ulkoista demodulaattoria audiosignaalin muuttamisessa digitaaliseen muotoon. Tällöin järjestelmä ottaa sarjaportin kautta vastaan sarjamuotoista bittivirtaa nopeudella, joka vastaa vastaanotettavan Baudot-signaalin nopeutta. Vaihtoehtoisesti järjestelmä voi käyttää esimerkiksi äänikorttia A/D-muuntimena, jolloin myös signaalin demodulointi ja dekodaus toteutetaan ohjelmallisesti.

Edelleen järjestelmän on pystyttävä ohjaamaan vastaanottoon käytettävää radiovastaanotinta siten, että kuunneltava taajuus voidaan asettaa tietokoneelta. Järjestelmän on kyettävä asettamaan myös muut asetukset, esimerkiksi vastaanottimen käyttämä lähetyslaji ja välitaajuussuodattimen päästökaistan leveys siten, että ne ovat soveltuvia radiokaukokirjoituksen vastaanottoon. Vastaanottimena käytetään Yaesu-Musen FT-920 -lähetinvastaanotinta, jota on mahdollista ohjata RS-232-sarjaliittymän kautta.

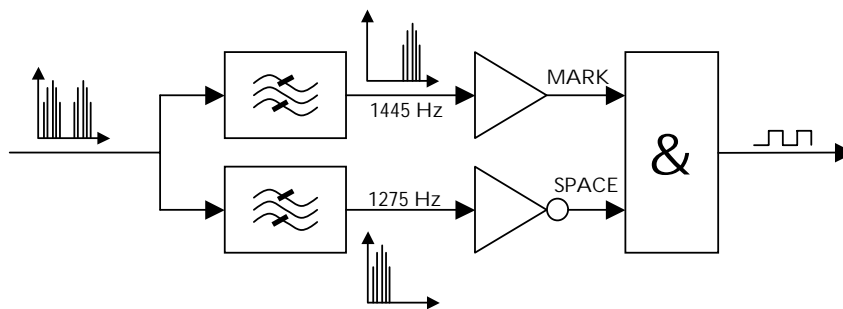
Tämän tutkielman tavoitteena on selvittää mahdollisuuksia kaukokirjoitus-signaalin vastaanoton automatisointiin. Toteutettavan järjestelmän kannalta tämä tarkoittaa sitä, että sen toteuttamisen yhteydessä tutkitaan menetelmiä järjestelmän parametrien automaattiseen asettamiseen vastaanotettavalle signaalille sopivaksi. Lisäksi selvitetään mahdollisuuksia kaukokirjoitinsignaalien hakemiseen ja tunnistamiseen valitulta taajuuskaistalta.

Järjestelmän toteutusympäristönä on Pentium-tasoisella prosessorilla ja SoundBlaster-äänikortilla varustettu PC-tietokone, jonka käyttöjärjestelmänä on GNU/Linux. Lisäksi tietokoneelta vaaditaan vähintään kaksi vapaata UART 16550 -yhteensopivaa sarjaporttia. Toista näistä sarjaporteista käytetään vastaanottimen ohjaamiseen ja toista mahdollisen ulkoisen demodulaattorin liittämiseen tietokoneeseen. Digitaaliseen signaalinkäsittelyyn perustuvassa järjestelmässä radiovastaanottimelta saatava äänisignaali syötetään joko äänikortin mikrofoniin tai linjatasoiseen sisäänmenoon, mikäli sellainen on käytävissä.

4. Ulkoista demodulaattoria käyttävä järjestelmä

4.1. Ulkoinen AFSK-demodulaattori

Yksinkertaisimmillaan ulkoinen demodulaattori vaihtotaajuusavainnusta var-
ten koostuu MARK- ja SPACE-signaalien äänitaajuuksille sovitetuista kaistan-
päästösuodattimista, joilla erotetaan tulevasta äänisignaalista halutut osat.
Suodattimilta saatava signaali muokataan esimerkiksi operaatiovahvistimilla ja
logiikkapiireillä binäärimuotoon siten, että MARK-signaali vastaa ykköstä ja
SPACE-signaali vastaavasti nollaa. Signaalitasot sovitetaan lopuksi tietokoneen
käyttämille jännitetasoille. Kuva 7 esittää demodulaattorin toimintaa lohkokaa-
viona.



Kuva 7. Audiosignaalin demodulointi

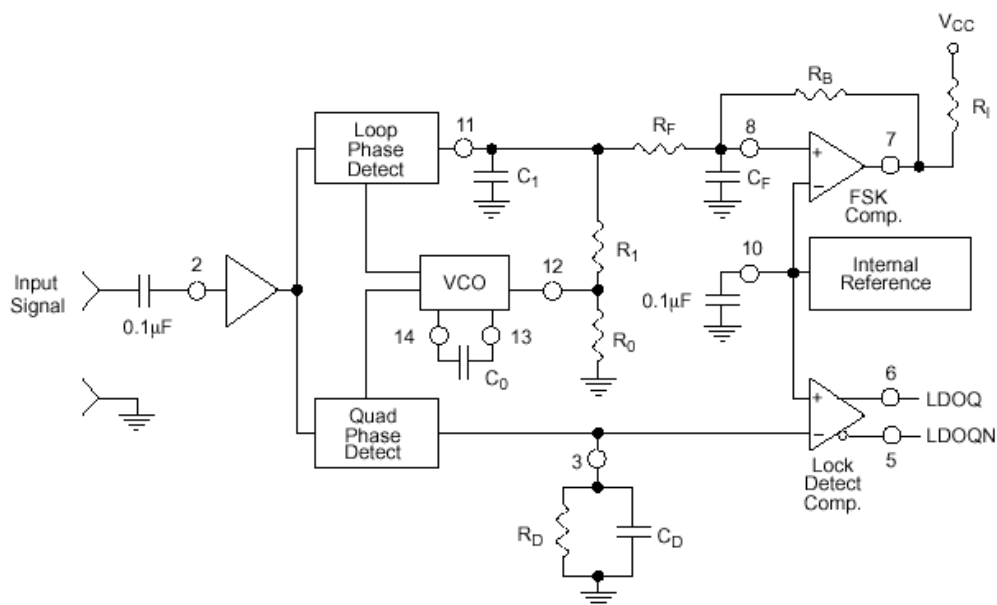
Kun kaistanpäästösuodattimet toteutetaan laadukkailla komponenteilla,
joiden arvot on optimoitu vastaanotettavan taajuusparin mukaisesti, saadaan
aikaiseksi demodulaattori, joka on varsin suorituskykyinen ja kykenee vas-
taanottamaan heikkoja kaukokirjoitussignaaleja melko häiriöisissäkin olosuh-
teissa. Erillisillä passiivisilla komponenteilla toteutettuna kytkentä on kuitenkin
melko monimutkainen ja kaistanpäästösuodattimien virittäminen vaatii tark-
koja mittalaitteita.

4.2. Demodulaattori modeempiirillä toteutettuna

AFSK-demodulaattori voidaan toteuttaa myös tähän tarkoitukseen suunnitel-
tua demodulaattoripiiriä käyttäen. Tällainen piiri on esimerkiksi Exarin XR-
2211, joka voidaan vähäisillä ulkoisilla komponenteilla sovittaa halutulle AFSK-
taajuusparille.

Kuvassa 8 on esitetty AFSK-demodulaattorin periaatteellinen kytkentäkaavio. Ulkoisille komponenteille lasketut optimaaliset arvot 170 hertsin ja 450 hertsin taajuussiirtymille on esitetty liitteessä 2. Arvot on laskettu XR-2211-piirin datalehdessä [Exar, 1992] esitettyjä kaavoja käyttäen. Euroopassa yleensä käytetään taajuusparia, jonka keskitaajuus on 1360 Hz. Yhdysvalloissa ja Japanissa taas suositaan korkeampaa taajuusparia, jonka keskitaajuus on 2210 Hz. Taajuussiirtymät ovat kuitenkin samat, eivätkä ne vaikuta kaukokirjoitus-signaalin vastaanottoon. Ainoastaan vastaanottimen esittämä radiotaajuus on toisen standardin mukaista signaalia vastaanotettaessa näiden keskitaajuuksien erotuksen verran poikkeava. Kaksisuuntaisissa lähetyksissä sekä lähettävän että vastaanottavan aseman tulisi käyttää saman standardin mukaisia laitteita, muutoin asemien lähetystaajuudet poikkeavat toisistaan. Liitteen 2 taulukkoon on laskettu komponenttiarvot kummankin standardin mukaisille äänitaajuuksille.

Valitut komponenttiarvot on pyörästetty lähimpiin R12-sarjan vakioarvoihin. Valitsemalla sama keskitaajuus kummallekin siirtymälle, pystytään suurin osa komponenttien arvoista pitämään samana molemmille keskitaajuuksille ja taajuussiirtymille. Potentiometrillä säädetään R_0 :n arvoa, joka vaikuttaa käytettävään keskitaajuuteen sekä taajuussiirtymään. Valitut arvot ovat jossakin määrin kompromisseja, mutta niillä saadaan aikaan riittävä suorituskyky haluttujen signaalien vastaanottoon. Mikäli kytkentää halutaan optimoida, on mahdollista korvata myös vastus R_1 potentiometrillä tai useammalla kiinteällä vastuksella. Nämä voidaan asettaa käyttöön esimerkiksi vaihtokytkimellä halutun taajuusparin vastaanottoa varten.



Kuva 8. Demodulaattori Exar XR-2211 -piirillä toteutettuna [Exar, 1992]

Demodulaattori liitetään tietokoneen sarjaporttiin. Audiosignaali syötetään demodulaattoriin nastaan 2 ja demoduloitu bittivirta saadaan nastasta 7. Tietokone vastaanottaa demodulaattorilta tulevan bittivirran RS-232-sarjaportin RxD-nastan kautta. Koska demodulaattorin käyttämät jännitetasot poikkeavat sarjaportin jännitetasoista, ne on sovitettava toisiaan vastaaviksi. Sovitus voidaan toteuttaa esimerkiksi Maximin MAX-202E-piirillä, joka muuttaa demodulaattoriin 0 - 12 voltin ns. TTL-tasot sarjaportin käyttämille -5 - +5 voltin jännitetasoille [Maxim, 1996]. Kytkeentään on lisätty myös piiri 7404, joka sisältää 6 loogista ei-piiriä. Ei-piireistä käytetään kahta: ensimmäisellä voidaan tarvittaessa kääntää signaalin polariteetti, toisella taas palautetaan kertaalleen käännetty signaali ohjaamaan valodiodia, joka osoittaa demodulaattorin lukitumisen AFSK-signaaliin. Toteutetun demodulaattorin kytkeentä ohjeiskomponentteineen on esitetty liitteessä 3.

4.3. Sarjaportin konfigurointi ja lukeminen

PC-tietokoneen sarjaporttia käytetään yleensä liikennöintiin oheislaitteiden tai muiden tietokoneiden kanssa. Tällöin käytetty nopeus on kohtalaisen suuri, tyypillisesti kymmeniä kilobaudeja. Koska radiokaukokirjoituksen vastaanotossa käytettävässä demodulaattorissa ei ole minkäänlaista puskuroidintia, vaan laitteen tuottama bittivirta vastaa vastaanotettavan signaalin nopeutta, täytyy sarjaportti konfiguroida nopeudelle sopivaksi.

RS-232-sarjaportin ohjelmallinen konfigurointi tapahtuu *sarjaliikennepiirin eli UART-piirin* (Universal Asynchronous Receiver Transmitter) rekistereitä käyttämällä. Halutut parametrit asetetaan UART-piirin muistialueelle. Kunkin sarjaportin (esim. COM1) rekistereille on varattu oma muistialue. UART 16550-sarjan piirin rekisterit on lueteltu taulukossa 1. Rekisterit on esitetty järjestyksessä piirin muistialueen alusta lukien. [Flyktman *et al.*, 1994]

Taulukko 1. 16550-yhteensopivan sarjaliikennepiirin rekisterit

Sijainti tavuina muistialueen alusta	Nimi	Selitys
0	TD/RD DIVLSB	Lähetettävä / vastaanotettava data. Nopeudenvaihtorekisterin vähemmän merkittävä tavu, mikäli LCR:n bitti d7 on asetettu.
1	IER DIVMSB	Keskeytysten ohjausrekisteri. Nopeudenvaihtorekisterin enemmän merkittävä tavu, mikäli LCR:n bitti d7 on asetettu.
2	IIR FIFO	Keskeytysten tunnistusrekisteri (vain luku). Sisäisten linjapuskureiden ohjaus.
3	LCR	Linjan ohjausrekisteri.
4	MCR	Oheislaitteen ohjausrekisteri.
5	LSR	Linjan tilarekisteri.
6	MSR	Oheislaitteen tilarekisteri
7	SCR	Ylimääräinen varastorekisteri

Sovitettaessa sarjaporttia vastaanotettavalle datalle, keskeinen UART:n rekisteri on *LCR-rekisteri* (Line Control Register). Rekisterillä asetetaan mm. tavun pituus, lopetusbittien määrä ja pariteettiasetukset. Rekisterin pituus on kahdeksan bittiä. Taulukossa 2 esitellään kunkin bitin merkitys [Engdahl, 1993].

Taulukko 2. LCR-rekisterin bittien käyttö

LCR-rekisterin bitti	Tarkoitus	Käyttö
d0 ja d1	Databittien lukumäärä	00 = 5 databittiä 01 = 6 databittiä 10 = 7 databittiä 00 = 8 databittiä
d2	Lopetusbittien lukumäärä	0 = 1 lopetusbitti 1 = 2 lopetusbittiä (1,5 jos 5 databittiä)
d3, d4 ja d5	Pariteettibitin käyttö	0xx = ei pariteettia 100 = pariton pariteetti (ODD) 101 = pariteetti aina 1 (MARK) 110 = parillinen pariteetti (EVEN) 111 = pariteetti aina 0 (SPACE)
d6	Lähetyksen ohjaus	0 = tavallinen lähetys 1 = asetetaan linja loogiseen nollaan
d7	Nopeuden valinta DLAB	0 = normaalitila 1 = asetetaan nopeuden valinta DIV-rekistereillä käyttöön

Linjanopeuden asettamiseen käytetään DIV-rekisteriä, joka on jaettu kahteen rekisteriin. Nämä DIVMSB- ja DIVLSB-rekisterit ovat normaalisti muussa käytössä, mutta ne voidaan ottaa nopeuden määrittämiseen asettamalla LCR-rekisterin bitti 7 ykköseksi. DIV-rekisterin arvo on jakoluku, joka määrittää linjanopeuden kaavalla

$$\text{jakoluku} = \frac{115000}{\text{linjanopeus}} .$$

Esimerkiksi linjanopeuksia 45,45 ja 50 baudia vastaavat jakoluvut ovat 2535 ja 2304. Taulukossa 3 on esitetty nämä jakoluvut purettuina enemmän (DIVMSB) ja vähemmän (DIVLSB) merkitseviin tavuihin.

Taulukko 3. Jakoluvut 45,45 ja 50 baudin nopeuksille

Nopeus (bps)	Jakoluku	Jakoluku binäärimuodossa		DIVMSB	DIVLSB
		MSB	LSB		
45,45	2535	00001001	11100111	9	231
50	2304	00001001	00000000	9	0

LCR-rekisterin bittejä käyttäen asetetaan myös muut vastaanotettavan datan ominaisuudet: 5 databittiä, 1,5 lopetusbittiä. Pariteettitarkastusta ei käytetä.

Monissa käyttöjärjestelmissä edellä esitetyt sarjaportin alustustoimenpiteet on toteutettu järjestelmän tarjoamina funktioina. Esimerkiksi POSIX-yhteensopivissa käyttöjärjestelmissä, kuten Linuxissa, alustaminen onnistuu varsin yksinkertaisilla funktiokutsuilla [Sweet, 1999]. Sarjaportti voidaan alustaa ohjelmakatkelmassa 1 esitetyllä funktiolla, joka palauttaa tiedosto-osoittimen käytävään porttiin.

Ohjelmakatkelman 1.

```
int open_port(char *port, int speed)
{
    int fd;
    struct termios settings;

    fd = open(port, O_RDWR | O_NOCTTY | O_NDELAY);
    if (fd == -1)
    {
        perror("Opening failed");
    }
    else
        fcntl(fd, F_SETFL, 0);
    tcgetattr(fd, &settings);
    cfsetispeed(&settings, speed);
    cfsetospeed(&settings, speed);
    settings.c_cflag &= ~PARENB;
    settings.c_cflag |= CSTOPB;
    settings.c_cflag &= ~CSIZE;
    settings.c_cflag |= CS8;
    settings.c_cflag &= ~OPOST;
    settings.c_lflag |= (ICANON | ECHO | ECHOE );

    tcsetattr(fd, TCSANOW, &settings);
    return(fd);
}
```

Tiedoston avaamisen jälkeen sitä voidaan lukea tiedosto-osoittimen kautta `read`-funktioilla. Ohjelmakatkelmassa 2 on esitetty funktio, joka lukee yhden merkin avatusta sarjaportista.

Ohjelmakatkelman 2.

```
char read_one_char(int fd)
{
    char ch;

    read(fd, &ch, 1);
    return(ch);
}
```

4.4. Baudot-koodin dekoodaaminen

Sarjaportin kautta vastaanotettava datavirta on edelleen Baudot-koodattua, siis 5-bittistä. Jotta vastaanotetut sanomat saataisiin luettavaan muotoon joko näyttöllä esittämistä tai tiedostoon tallentamista varten, on datalle suoritettava muunnos ASCII-koodiin. Baudot-koodi on esitetty liitteessä 1.

Baudot-koodissa kullakin merkillä on kaksi merkitystä. Se voi olla joko kirjain tai muu merkki. Tulkinta riippuu siitä, onko vastaanottohetkellä kirjaintaso vai numerotaso käytössä. Tason vaihto tapahtuu omalla merkillään, joka sisältyy datavirtaan. Kummallekin tasolle on oma merkkinsä. Taulukossa 4 on esitetty merkkijonon "OH7KA" muodostaminen Baudot-koodilla.

Taulukko 4. Sekä kirjaimia että numeroita sisältävän merkkijonon esitys Baudot-koodilla.

'O'	'H'	[FIGS]	'7'	[LTRS]	'K'	'A'
11000	10100	11011	00111	11111	01111	00011

Jokaisen vastaanotetun merkin osalta joudutaan tekemään siis valinta, tulkitanko se kirjaimeksi, numerotason merkiksi vai onko kyseessä tasonvaihtomerkki, joka vaikuttaa sitä seuraaviin merkkeihin. Dekoodaus toteutetaan ohjelmakatkelmassa 3 esitettynä funktiona, joka palauttaa vastaanotettua koodia vastaavan merkin tai suorittaa tasonvaihdon, mikäli vastaanotettu merkki tulkitaan tasonvaihtomerkiksi.

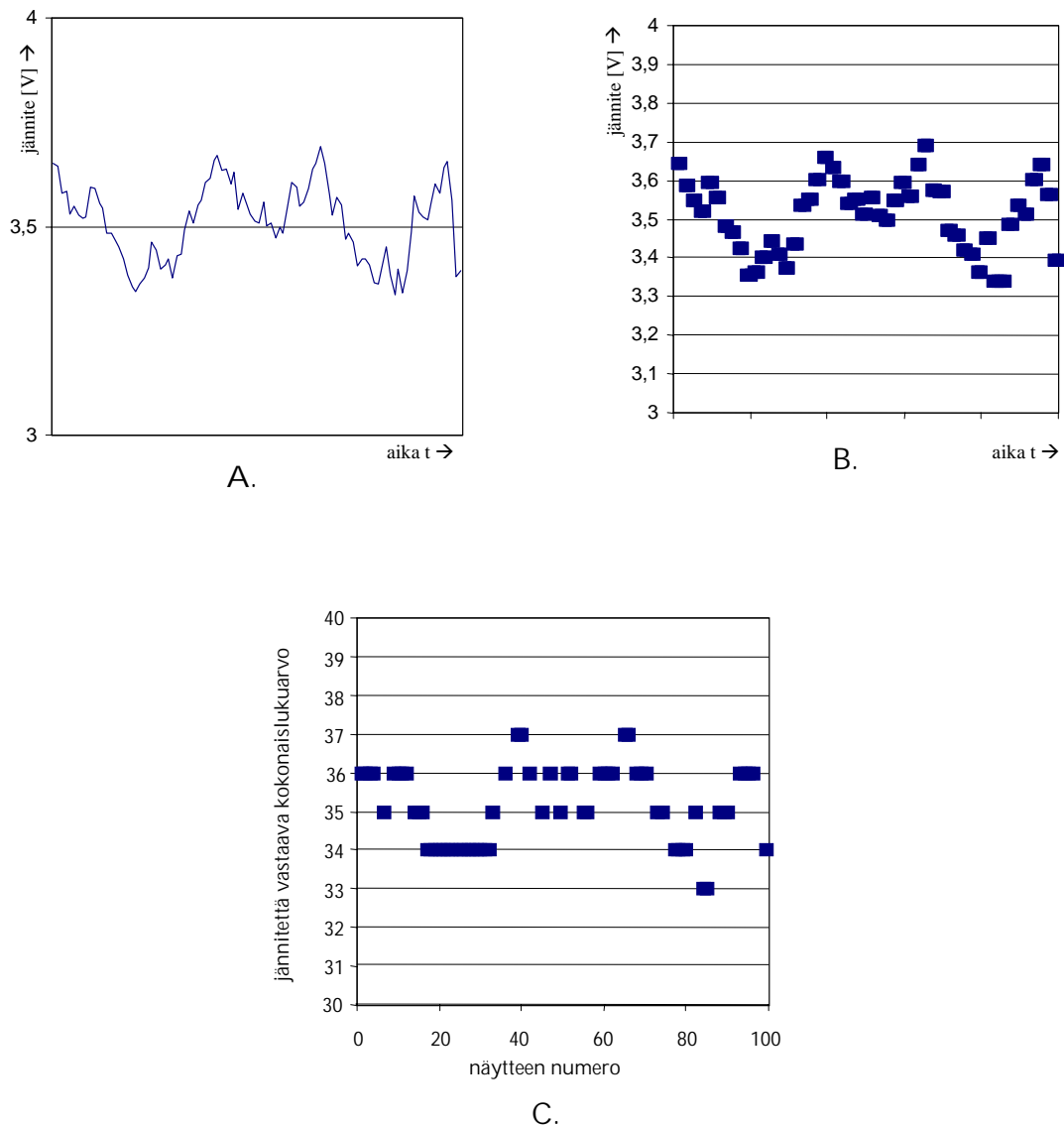
Ohjelmakatelma 3.

```
int baudot(unsigned char data) /* Baudot-dekooderi*/
{
    static int mode=0;
    static char letters[32]={ '<', 'E', '\n', 'A', ' ', 'S', 'I', 'U',
                             '\n', 'D', 'R', 'J', 'N', 'F', 'C', 'K',
                             'T', 'Z', 'L', 'W', 'H', 'Y', 'P', 'Q',
                             'O', 'B', 'G', '^', 'M', 'X', 'V', '^' };
    static char symbols[32]={ '<', '3', '\n', '-', ' ', ' ', ' ', '8', '7',
                              '\n', '$', '4', '#', ' ', ' ', 'Å', ':', '(',
                              '5', '+', ')', '2', 'Ö', '6', '0', '1',
                              '9', '?', 'Ä', '^', '.', '/', '=', '^' };

    int d;
    d=0;
    switch (data) {
    case 0x1f : mode=0; break; /*
    case 0x1b : mode=1; break;
    default: if (mode==0) d=letters[data];
             else d=symbols[data]; break;
    }
    return d;
}
```

5. Digitaaliseen signaalinkäsittelyyn perustuva järjestelmä

Tyypillinen *digitaalista signaalinkäsittelyä* (DSP, digital signal processing) hyödyntävä sovellus sisältää useita vaiheita. Ensiksi jatkuva-aikaisesta analogisesta signaalista otetaan näytteitä tietyllä *näytteenottotaajuudella* (sampling frequency). Tämän vaiheen tuloksena käytettävissä on analogisesta signaalista määrävälein otettuja mittaustuloksia eli jatkuva-aikainen signaali on muutettu diskreettiaikaiseksi.



Kuva 9. Analoginen (A), näytteistetty (B) ja kvantisoitu signaali (C)

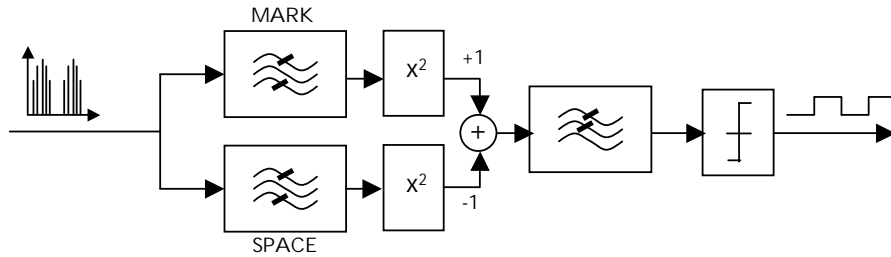
Esimerkiksi kuvan 9 kaaviossa A analogisesta signaalista, tässä tapauksessa jännitteestä, saadaan kaavion B mukaiset näytteet. Seuraavaksi diskreetti aikaiset näytteet kvantisoidaan eli mitatut arvot sijoitetaan A/D-muuntimen käyttämälle asteikolle. Kuvan 9 kaaviossa C esimerkin A/D-muunnin pystyy esittämään jännitearvoja väliltä 0 – 25,5 voltia kokonaisluvulla nollan ja 255:n väliltä. Näin ollen jännitteet pystytään esittämään 0,1 voltin tarkkuudella ja esimerkiksi 3,422 voltin jännite esitetään kokonaisluvulla 34. A/D-muunnoksesta siis seuraa *kvantisointivirhe* (quantization error), jonka suuruus määräytyy näytteen bittien lukumäärän mukaan [Smith, 1999]. Kvantisointivirheen vaikutus esiintyy yleensä lisääntyneenä kohinana, jota kutsutaan *kvantisointikohinaksi* (quantization noise) [Carlson *et al.*, 2002].

Kun analoginen signaali on saatu digitaaliseen ja diskreetti aikaiseen muotoon, sitä voidaan muokata. Siitä voidaan esimerkiksi poistaa kohinaa siten, että varsinainen signaali säilyy mahdollisimman hyvin tai erotella signaalissa olevia mielenkiintoisia piirteitä. Tätä vaihetta kutsutaan suodattamiseksi. Suodatuksen jälkeen signaali voidaan esimerkiksi muuntaa takaisin analogiseksi D/A-muuntimella tai käsitellä muulla tavalla. Kaukokirjoituksen vastaanottojärjestelmässä tavoitteena on saada signaali dekodattua luettavaan muotoon.

5.1. DSP-pohjaisen järjestelmän rakenne

Digitaaliseen signaalinkäsittelyyn perustuvat menetelmät ovat yleistyneet viime vuosina monien muiden sovellusten ohella myös radiokaukokirjoituksen vastaanotossa. Ensimmäiset tällaiset järjestelmät käyttivät audiosignaalin käsittelyyn tarkoitettuja DSP-proessoreita [Forrer, 1996]. Näissä keskityttiin lähinnä aikaisemmin passiivisilla komponenteilla tai modeemipiireillä toteutettujen äänitaajuussuodattimien korvaamiseen ohjelmallisesti toteutetuilla suodattimilla.

Digitaalista signaalinkäsittelyä käyttäen voidaan myös demodulaattorin tehtävät antaa tietokoneen suoritettavaksi. Tällöin äänisignaali syötetään suoraan vastaanottimesta tietokoneeseen esimerkiksi äänikortin sisäänmenon kautta. Ulkoista demodulaattoria ei tällöin tarvita, joten liitäntä on hyvin yksinkertainen. Tässä esiteltävän järjestelmän signaalinkäsittely perustuu espanjalaisen Jesús Ariasin Linux-ympäristöön toteuttamaan RTTY-ohjelmaan [Arias, 2001]. Kyseinen ohjelma on ns. GPL-lisenssin [GPL, 1991] piirissä ja siten vapaasti lähdekoodeineen saatavissa ja edelleen kehitettävissä. Signaalinkäsittelyosuuden lohko kaavio on esitetty kuvassa 10.



Kuva 10. Lohkokaavio kaukokirjoitussignaalin käsittelystä

Kuten ulkoisella demodulaattorilla toteutetussa järjestelmässäkin, äänisignaalista erotetaan MARK- ja SPACE-taajuudet kaistanpäästösuodattimilla, jotka tällä kertaa toteutetaan ohjelmallisesti. Kaistanpäästösuodattimilta saaduista signaaleista erotetaan neliöimällä niiden amplitudit. Laskemalla näiden amplitudien erotus, saadaan tuloksena joko positiivisia tai negatiivisia lukuja, riippuen siitä onko kyseessä MARK- vai SPACE-signaali.

5.2. Audiosignaalin lukeminen

Vastaanottimesta saatava audiosignaali syötetään tietokoneen äänikortin mikrofoniiliittimeen tai linjatasoiseen sisääntulokanavaan, mikäli sellainen on käytettävissä. Liitännässä on suotavaa käyttää erotusmuuntajaa tai impedanssisovitinta, jolloin radio ja äänikortti eivät ole galvaanisessa yhteydessä toisiinsa. Näin vältetään laitteiden eri maapotentiaaleista johtuvat ongelmat ja jopa laitevauriot.

Linux-ympäristössä äänikortti, kuten muutkin laitteet, näkyy *laitetiedostona* (device file). Äänikortin tapauksessa käytettävissä on vaihtoehtoisia laitetiedostoja, joissa käytetään erilaisia koodauksia. Kaukokirjoitussignaalin lukemiseen sopii esimerkiksi lineaarista koodausta käyttävä laitetiedosto `/dev/dsp`. Tiedoston avaamisen jälkeen äänikortti alustetaan tiedostokuvaajan kautta `ioctl`-funktiolla, jolla voidaan määritellä näytteenottotaajuus ja muut tarvittavat parametrit [Tranter, 2001].

Shannonin näytteenottoteoreeman mukaan analoginen signaali voidaan muodostaa uudelleen näytteenottoaajuudeltaan, jos näytteenottotaajuus on vähintään kaksi kertaa niin suuri kuin suurin signaalin sisältämä taajuuskomponentti. Korkeinta taajuutta, jota pystytään tietyllä näytteenottotaajuudella esittämään, kutsutaan Nyquistin taajuudeksi. [Lynn and Fuerst, 1998] Radiokaukokirjoituksen vastaanottoon käytettävän yksisivunauhapuhelähetteen vastaanottamiseen tarkoitettujen vastaanottimien kaistanleveys rajoittaa äänisignaalien ylärajan alle 3000 hertsiin, joten näytteenottotaajuudeksi voidaan valita sopiva taajuus 6000 hertsin yläpuolelta. Useimmat äänikortit tukevat esimerkiksi 11025 hertsin

näytteenottotaajuutta. Kukin näyte on 8 bitin kokoinen ja näytteet luetaan 128 tavun kokoisissa osioissa puskuriiin.

Näytteiden lukeminen äänikortilta tapahtuu `read`-funktiolla. Funktio `get_audio_sample`, jonka toteutus on esitelty ohjelmakatkelmassa 4, lukee yhden näytteen ja palauttaa sitä vastaavan liukulukuarvon. Mikäli funktiota kutsuttaessa puskurissa on palauttamattomia arvoja, ei uutta näytettä lueta äänikortilta, vaan palautetaan puskurista seuraava arvo.

Ohjelmakatkelma 4.

```
#define NBUF 128

float get_audio_sample()
{
    static int bp=0, ns=0;
    static unsigned char audio_buf[NBUF];

    if (bp==ns) {
        ns=read(audio_fd, audio_buf, NBUF);
        bp=0;
    }
    return (float)(audio_buf[bp++]-128);
}
```

5.3. Suodattimien toteuttaminen ohjelmallisesti

Suodattimien tehtävänä on poistaa vastaanotetusta signaalista äänitaajuudet, jotka eivät sisällä informaatiota. Kuten edellä on esitetty, kaukokirjoitussignaaleissa tietoa siirretään kahta äänitaajuutta, MARK- ja SPACE-taajuuksia käyttäen. Vastaanotetussa muutaman kilohertsin levyisessä äänikaistassa on näin ollen huomattavan paljon taajuuksia, jotka ovat vastaanotettavan signaalin kannalta turhia.

Digitaalisesti toteutettavia suodattimia suunniteltaessa mallinnetaan vastaavien analogisten suodattimien toimintaa. Digitaalisilla suodattimilla on joi-takin etuja analogisiin suodattimiin nähden. Esimerkiksi ympäristön fysikaaliset muutokset, kuten lämpötilan vaihtelut, eivät vaikuta suodattimen toimintaan, koska suodattimen parametrit pysyvät vakioina. Toisaalta saman suodattimen ominaisuudet ovat helposti muutettavissa erilaisia signaaleja varten. Tätä voidaan kaukokirjoituksen vastaanottojärjestelmässä soveltaa esimerkiksi erilaisia taajuuspareja käyttävien signaalien vastaanottamiseen samalla järjestelmällä. Analogisessa järjestelmässä tämä vaatisi komponenttien säätämistä ja samalla suodattimien tarkkuus ja stabiilisuus saattaisi kärsiä. Digitaalisesti voidaan myös toteuttaa monimutkaisia suodatinalgoritmeja, joita ei olisi mahdollista toteuttaa järkevästi analogisin menetelmin. Rajana ovat lähinnä käytettävien prosessoreiden teho ja mahdolliset aikavaatimukset. Lisäksi oh-

jelmallisesti toteutettujen DSP-suodattimien monistaminen on helppoa ja halpaa, mikä on merkityksellistä kaupallisia sovelluksia ajatellen.

Suodattimien suunnittelussa on kyse sopivan digitaalisen suodatintyyppin valinnasta ja sopivien parametrien, siis suodatinkertoimien, määrittelystä. Kaukokirjoitusjärjestelmää varten MARK- ja SPACE-taajuuksille tarvittavat kaistanpäästösuodattimet voidaan toteuttaa rekursiivisina suodattimina. Näistä saatava signaali riippuu syötetyn signaalin lisäksi myös aikaisempien näytteiden tuottamista arvoista. Tällöin suodattimen määrittelyyn tarvittavien kertoimien määrä pysyy varsin pienenä. Toisaalta huonosti määritellyt kertoimet saattavat tehdä suodattimesta epävakaa [Lynn and Fuerst, 1998]. Rekursiiviset suodattimet ovat usein *IIR-tyyppisiä* (Infinite Impulse Response), eli teoreettisesti niiden vaste yksittäiselle impulssillekin on äärettömän kestoinen. Analoginen suodatin voidaan mallintaa IIR-suodattimeksi käyttämällä bilineaari-muunnosta.

5.3.1. Suodattimien kertoimien määrittelystä

Edellä mainitut kertoimet määrittelevät suodattimen ominaisuudet, esimerkiksi päästö- ja estokaistojen alku- ja lopputaajuudet. Jotta suodatin pysyisi vakaana, on syytä pitää käytettävien kertoimien määrä pienenä. Kertoimien määrittelyyn on olemassa valmiita ohjelmia. Esimerkiksi Tony Fisherin Yorkin yliopistossa kehittämää mkfilter-ohjelmaa on mahdollista käyttää internetin kautta web-sovelluksena [Fisher, 1999]. Myös esimerkiksi Matlab-ohjelmistolle on olemassa valmiita sovelluksia erilaisten suodattimien määrittelyyn. Yleinen algoritmi rekursiiviselle IIR-suodattimelle on muotoa

$$y_0 = a_0 x_i + a_1 x_{i-1} + \dots + a_n x_{i-n} + b_1 y_{i-1} + \dots + b_m y_{i-m}. \quad (5.3.1)$$

Kaavassa (5.3.1) esiintyvä x_i on viimeisin näyte äänisignaalista, y_{i-1} vastavasti sitä edeltävä suodattimelta saatu palautusarvo. Näin ollen edelliset näytteet vaikuttavat suodattimen palauttamiin arvoihin ja takaisinkytkentä toteutuu. Kertoimilla a ja b määritellään suodattimen ominaisuudet.

Koska kaukokirjoituksen vastaanottojärjestelmässä saattaa olla tarpeen koikeilla erilaisia AFSK-taajuuspareja, olisi toivottavaa että näitä voitaisiin muuttaa ohjelman suorituksen aikana. Koska tällöin joudutaan myös muuttamaan digitaalisten suodattimien päästökaistoja, joudutaan näille laskemaan kertoimet uudelleen. Seuraavassa esitetään kaistanpäästösuodattimen kertoimien määrittelyyn tarvittavat kaavat sekä c-kielinen funktio kertoimien laskemiseksi Jesús Ariasin toteutuksen mukaisesti [Arias, 2001].

Haluttu suodattimen päästökaistan keskitaajuus f_0 muunnetaan radiaanimuotoon resonanssitaajuudeksi ω_0 kaavalla

$$\omega_0 = 2\pi f_0.$$

Koska näytteistä koottu signaali on luonteeltaan diskreettiä, syntyy bilineaarimuunnoksessa jatkuva-aikaiseen signaaliin nähden poikkeama. Tämä poikkeama on kompensoitava *esikorjauksella* (prewarping). Tähän käytetään kaavaa

$$\hat{\omega}_0 = 2f_s \tan\left(\frac{\omega_0}{2f_s}\right). \quad (5.3.2)$$

Esikorjauksessa otetaan huomioon näytteenottotaajuuden f_s vaikutus [Jackson, 1986]. Jatkossa suodattimien kertoimien laskemiseen käytetään kaavan (5.3.2) tuloksena saatavaa resonanssitaajuutta $\hat{\omega}_0$.

Tarvittava siirtofunktio [Arias, 2001] jatkuva-aikaiselle signaalille Laplace-muunnoksen termein esitettynä on

$$H(s) = \frac{\frac{1}{Q} \frac{s}{\hat{\omega}_0}}{\frac{s^2}{\hat{\omega}_0^2} + \frac{1}{Q} \frac{s}{\hat{\omega}_0} + 1}. \quad (5.3.3)$$

Kaavassa (5.3.3) esiintyvä s on ns. *Laplace-muuttuja* ja Q on suodattimelle haluttu *hyvyysarvo* (quality factor). Kaistanpäästösuodattimen hyvyysarvo voidaan määritellä esimerkiksi suodattimen päästökaistan keskitaajuuden ja kaistanleveyden välisenä suhteena [Lane and Hillman, 1992].

Siirtofunktio muunnetaan bilineaarimuunnoksella z-tasoon, jolloin tuloksena saadaan diskreettiaikainen vastine kaavassa 5.3.3 esitellylle siirtofunktiolle. Bilineaarimuunnoksessa kaavan 5.3.3 tekijä s korvataan vastaavalla z-tason esityksellä [Smith, 1999]:

$$s = 2f_s \frac{1 - z^{-1}}{1 + z^{-1}}.$$

Esityksen yksinkertaistamiseksi bilineaarimuunnoksella saadussa siirtofunktiossa käytetään tekijää α kuvaamaan näytteenottotaajuuden ja resonanssitaajuuden suhdetta. Tekijä α määritellään siis kaavalla

$$\alpha = \frac{f_s}{\hat{\omega}_0}.$$

Bilineaarimuunnoksen tuloksena saadaan siirtofunktio, joka Z-tason termein esitettynä on muotoa

$$H(z) = \frac{\frac{2\alpha}{Q}(1-z^{-2})}{(4\alpha^2 + \frac{2\alpha}{Q} + 1) + (2 - 8\alpha^2)z^{-1} + (4\alpha^2 - \frac{2\alpha}{Q} + 1)z^{-2}}. \quad (5.3.4)$$

Siirtofunktiossa (5.3.4) esiintyvä tekijä z^{-n} voidaan ajatella aikatason esityksessä vastaavan n :nen näytteenottosyklin viivästystä. Tekijästä z^{-n} käytetäänkin nimitystä *viivästystekijä* (delay operator), vastaava *aikaistustekijä* (advance operator) olisi z^n . [Jackson, 1986]

Signaalista voidaan erottaa Z-tason esitykset tulosignaali $X[z]$ ja lähtösignaalille $Y[z]$ kaavalla

$$H(z) = \frac{Y(z)}{X(z)}.$$

Siirtofunktio palautetaan nyt algoritmisesti toteutettavaan muotoon, jota voidaan soveltaa käsiteltäville signaaleille:

$$y_i = \frac{\frac{2\alpha}{Q}}{4\alpha^2 + \frac{2\alpha}{Q} + 1}(x_i - x_{i-2}) + \frac{8\alpha^2 - 2}{4\alpha^2 + \frac{2\alpha}{Q} + 1}y_{i-1} - \frac{4\alpha^2 - \frac{2\alpha}{Q} + 1}{4\alpha^2 + \frac{2\alpha}{Q} + 1}y_{i-2}. \quad (5.3.5)$$

Kaava (5.3.5) voidaan edelleen esittää alkuperäistä rekursiivisten IIR-suodattimien algoritmia 5.3.1 vastaavassa muodossa, jolloin saadaan seuraavat kaistanpäästösuodattimen kertoimien määrittämiseen tarvittavat kaavat:

$$a_0 = -a_2 = \frac{\frac{2\alpha}{Q}}{4\alpha^2 + \frac{2\alpha}{Q} + 1}, \quad (5.3.6)$$

$$b_1 = \frac{8\alpha^2 - 2}{4\alpha^2 + \frac{2\alpha}{Q} + 1} \quad (5.3.7)$$

ja

$$b_2 = -\frac{4\alpha^2 - \frac{2\alpha}{Q} + 1}{4\alpha^2 + \frac{2\alpha}{Q} + 1}. \quad (5.3.8)$$

Ohjelmakatkelmassa 5 on esitetty funktio, joka laskee kaistanpäästösuodattimen kertoimet liukulukuvektoriin `pcoef` edellä esitettyjen kaavojen (5.3.6, 5.3.7 ja 5.3.8) mukaisesti. Funktion parametreiksi tarvitaan kaistanpäästösuodattimen keskitaajuus f_0 , sekä hyvyyskerroin Q . Kaukokirjoitusjärjestelmässä tarvittavan alipäästösuodattimen kertoimet määritellään vastaavalla tavalla, mutta käytettävä siirtofunktio on hieman erilainen.

Ohjelmakatkelma 5.

```
#define FS    11025    /* naytteenottotaajuus */
void gen_coef_bandpass(float f0, float Q, float *pcoef)
{
    float w0, a, d;
    w0=2.0*M_PI*f0;
    /* esikorjaus */
    w0=2.0*FS*tan(w0/FS/2.0);
    a=FS/w0;
    d=4.0*a*a + 2.0*a/Q + 1.0;
    (*pcoef++)=1.0/d
    (*pcoef++)=(8.0*a*a - 2.0 )/d;
    (*pcoef++)=-(4.0*a*a - 2.0*a/Q + 1.0)/d;
}
```

Itse kaistanpäästösuodatin toteutetaan kaavan 5.3.5 mukaisesti. Kun suodattinta käytetään signaalinäytteen x_i käsittelyyn, sille annetaan näytteen lisäksi syötteeksi suodattimelle määritellyt kertoimet sekä edellisten näytteiden palauttamien arvot, jotka on kerätty liukulukuvektoreihin. Funktion c-kielinen toteutus on esitetty ohjelmakatkelmassa 6.

Ohjelmakatkkelma 6.

```
extern inline float biquad_bandpass(float x, float *pcoef, float *buf)
{
    float y;
    y =(*pcoef++)*(x-buf[1]);
    y+=(*pcoef++)*buf[2];
    y+=(*pcoef++)*buf[3];

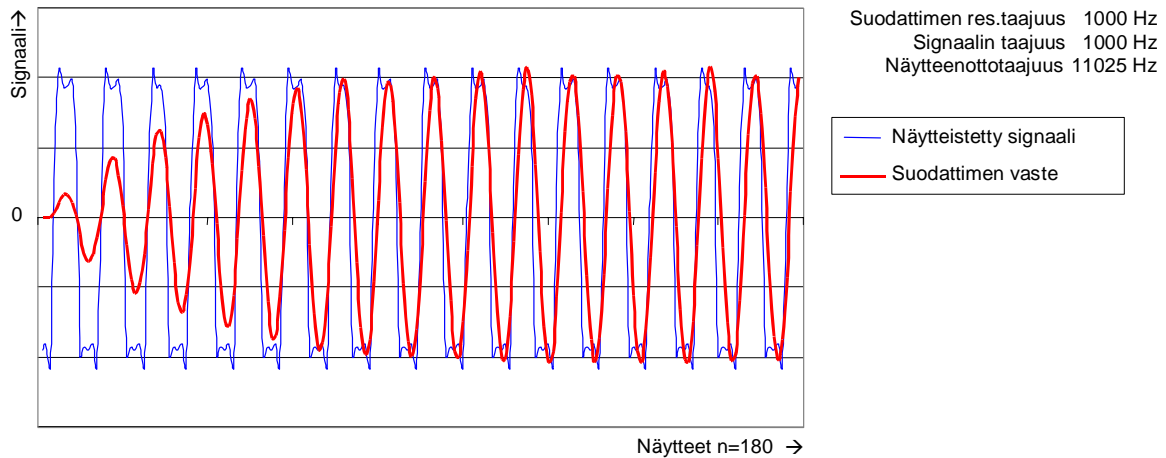
    buf[1]=buf[0]; buf[0]=x;
    buf[3]=buf[2]; buf[2]=y;

    return y;
}
```

5.3.2. Kaistanpäästösuodattimen kokeilua

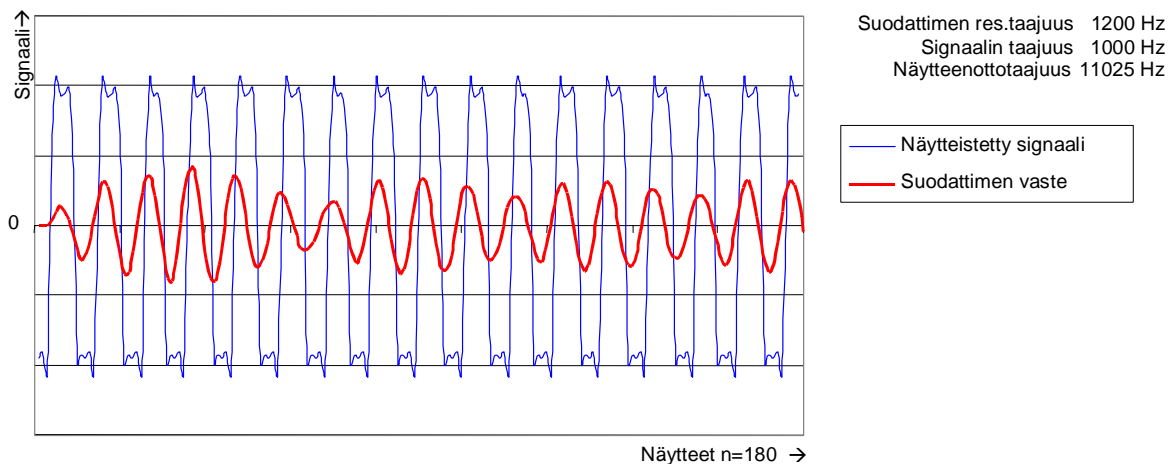
Seuraavaksi tutkitaan DSP-suodattimen toimintaa käytännön kokeella, mallintamalla suodatin taulukkolaskentaohjelmassa. Tietokoneen äänikortin sisäänmenoon syötetään äänigeneraattorista 1000 hertsin siniaaltomuotoinen äänisignaali. Signaalinkäsittelyohjelma kerää 180 peräkkäistä näytettä kohdassa 5.2 esitellyllä menetelmällä. Näytteenottotaajuudeksi on asetettu 11025 hertsiä. Näytteet siirretään taulukkolaskentaohjelmaan, jonne on määritelty kaistanpäästösuodattimen toteuttava algoritmi kaavan 5.3.5 mukaisesti. Taulukkolaskentaohjelmassa lasketaan suodattimen kertoimet halutulle päästökaistan keskitaajuudelle ja ohjelma laskee kutakin näytettä vastaavan suodattimen palautusarvon.

Kuvien 11 ja 12 sininen käyrä esittää näytteiden muodostamaa aaltomuotoa. Punainen käyrä kertoo suodattimen palauttaman arvon kullekin näytteelle. Kuvassa 11 suodattimen kertoimet on laskettu 1000 hertsin päästökaistan keskitaajuudelle, toisin sanoen tutkittava signaali on keskellä päästökaistaa. Kuvasta nähdään, että suodattimen vaste mukautuu syötettyyn signaaliin varsin hyvin jo noin sadan ensimmäisen näytteen jälkeen.



Kuva 11. Kaistanpäästösuodattimen vaste resonanssitaajuudellaan

Kuvan 12 tapauksessa suodattimen kertoimet on laskettu siten, että suodattimen päästökaistan keskitaajuus on 1200 hertsiä, eli 200 hertsiä käytettävän testisignaalin yläpuolella. Tällöin suodattimen palauttama vaste poikkeaa huomattavasti testisignaalista, eikä resonanssia löydy. Mitä suurempi ero suodattimen päästötaajuuden ja syötettävän signaalin taajuuden välillä on, sitä pienempi on signaalin ja suodattimen vasteen välinen korrelaatio.



Kuva 12. Esimerkki kaistanpäästösuodattimen vasteesta resonanssitaajuuden ulkopuolella

5.4. Signaalin demodulointi

Demodulointivaiheessa käytetään edellä esiteltyjä suodattimia, joten varsinainen demodulointifunktio näyttää varsin yksinkertaiselta. Äänisignaalista otettu näyte syötetään MARK- ja SPACE-taajuuksia vastaaviin kaistanpäästösuodat-

timiin, joilta saadut arvot neliöidään kyseisten taajuuksien *amplitudien tehollisarvon* (RMS, Root Mean Square) muodostamiseksi. Amplitudien erotukselle suoritetaan vielä kaksinkertainen alipäästösuodatus.

Ohjelmakatkelmassa 7 esitetty funktio palauttaa arvon 1, mikäli äänisignaalista otettu näyte sisältää MARK-taajuuteen sopivia komponentteja, muussa tapauksessa funktio palauttaa arvon 0.

Ohjelmakatkelma 7.

```
int demodulator()
{
    float sample, xspace, xmark, y;

    sample=get_audio_sample();
    xspace=biquad_bandpass(sample,bp0_c,bp0_b);
    xmark=biquad_bandpass(sample,bp1_c,bp1_b);
    xspace*=xspace; /* RMS */
    xmark*=xmark; /* RMS */
    y=biquad_lowpass(biquad_lowpass(xmark-xspace,lp1_c,lp1_b)
                    ,lp2_c,lp2_b);

    return (y>0.0)?1:0;
}
```

5.5. Dekoodaaminen

Dekoodausvaihe voidaan jakaa useampaan osaan. Ensiksi signaalivirrasta tunnistetaan yksittäisiä bittejä, sitten biteistä kootaan kokonaisia tavuja. Tavut dekodataan lopulta kirjainmerkeiksi.

Yksittäisten bittien tunnistaminen tapahtuu vertaamalla peräkkäisiä demodulaattorilta yhdestä näytteestä saatuja arvoja. Näytteiden synkronointi signaalin symbolivirtaan voidaan toteuttaa *vaihelukittuna silmukkana* (PLL, phase locked loop) [Carlson *et al.*, 2002]. Tässä järjestelmässä PLL on toteutettu digitaalisesti siten, että silmukan muodostaa staattinen 16-bittinen kokonaisluku-muuttuja, joka ylivuototilanteessa aiheuttaa bittiarvon tulostumisen. Kutakin demodulaattorilta saatua näytettä kohden muuttujaa kasvatetaan vastaanotettavan signaalin symbolinopeutta ja näytteenottotaajuutta vastaavalla arvolla. Esimerkiksi 50 baudin nopeudella yhden symbolin, siis bitin, kesto on 20 millisekuntia. Jos näytteenottotaajuus on 11025 hertsiä, saadaan kutakin symbolia kohden 220,5 näytettä. Koska yhden bitin kesto vastaa muuttujan maksimiarvoa 2^{16} , saadaan yhtä näytettä vastaava arvo seuraavasti:

$$\frac{\text{symbolinopeus} * 2^{16}}{\text{näytteenottotaajuus}} \quad (5.5.1)$$

Kun nopeus on 50 baudia, saadaan kaavalla (5.5.1) yhtä näytettä vastaavaksi arvoksi 297. Muuttujaa kasvatetaan tällä arvolla jokaista demodulaattorilta saatua näytettä kohden. Kun muuttuja vuotaa yli, palautetaan viimeksi demodulaattorilta saatua arvoa vastaava bitti. Jotta synkronointi toteutuisi, pyritään bitin vaihtuminen seuraavaksi, eli demodulaattorin ilmoittama signaalitason muuttuminen, ajoittamaan silmukan keskivaiheille. Mikäli taso vaihtuu ennen kuin muuttuja on saavuttanut puolet maksimiarvostaan, aikaistetaan bitin tulostumista lisäämällä muuttujaan 1/32 maksimiarvosta. Vastaavasti mikäli puoliväli on ohitettu ennen tason muuttumista, viivästetään silmukkaa vähentämällä muuttujasta 1/32 maksimiarvosta [Arias, 2001]. Silmukan toteuttama funktio on esitetty ohjelmakatkelmassa 8. Funktio palauttaa demodulaattorilta luettua signaalia vastaavan tason palautusarvonsa ylimpänä bittinä.

Ohjelmakatkelman 8.

```
int get_bit_raw()
{
    static int x0;
    int x,f;

    for (f=0;;){
        x=demodulator();
        if ((x^x0)) { /* XOR edellisen näytteen kanssa */
            if (!f) {
                if (bit_phase<0x8000) bit_phase+=2048; /* aik */
                else bit_phase-=2048; /* myöh */
                f=1;
            }
        }
        x0=x;
        x=bit_phase;
        bit_phase+=fskdp->phs;
        if ((x&0x8000) && !(bit_phase&0x8000)) break;
    }

    return x0<<7;
}
```

Jotta biteistä voidaan koota tavuja, täytyy ensiksi löytää tavun aloitusbitti. Tässä voidaan käyttää hyväksi tietoa siitä, että lopetusbitti on kestoiltoistakertainen muihin bitteihin nähden ja sitä seuraava seuraavan tavun aloitusbitti on vastakkaisella tasolla lopetusbittiin verrattuna. Ohjelmakatkelmassa 9 on esitetty silmukka, joka tunnistaa signaalin laskevan reunan ja mikäli vielä puolen bitin keston jälkeen signaali on SPACE-tasolla, oletetaan vastaanotettu bitti aloitusbitiksi.

Ohjelmakatkkelma 9.

```
/* Aloitusbitin tunnistaminen */
do {
    /* Signaalin laskevan reunan tunnistaminen */
    while (!demodulator());
    while (demodulator());
    /* odotetaan 0.5 bitin verran */
    for(i=0;i<0x8000;i+=fskdp->phs) x1=demodulator();
} while (x1); /* Toistetaan kunnes aloitusbitti loytyy */
```

Varsinaiset databitit tunnistetaan ohjelmakatkkelmassa 8 esitellyllä funktiolla ja luetaan yksi kerrallaan muodostettavaa merkkiä vastaavaan tavuun. Samoin luetaan ja tarkastetaan pariteettibitti, mikäli pariteetti on käytössä. Lopuksi luetaan lopetusbitti, jonka Baudot-koodissa pitäisi olla ykkönen. Mikäli näin ei ole, on vastaanotetussa tavussa virhe, joka voidaan osoittaa esimerkiksi tulostamalla erikoismerkki.

Näin muodostettu tavu muunnetaan vastaavaksi kirjain- tai numeromeriksi ohjelmakatkkelman 3 mukaisella funktiolla, joka esiteltiin kohdassa 4.4.

6. Vastaanoton automatisoinnista

Yksi tämän tutkielman tavoitteista on tutkia mahdollisuuksia radiokaukokirjoituksen vastaanoton automatisointiin. Lähinnä kyseeseen tulee vastaanottimen ohjaaminen kaukokirjoitusjärjestelmästä käsin sekä vastaanottoparametrien automaattinen asettaminen seurattavan signaalin perusteella

Kaukokirjoituksen vastaanotossa on ollut yleisenä ongelmana vastaanottimen asettaminen oikealle taajuudelle. Ihminen oppii tekemään tämän ajan mittaan korvakuulolta, mutta myös erilaisia visuaalisia viritysilmaisimia on myös käytetty. Esimerkiksi vastaanotetun signaalin MARK- ja SPACE-komponentit voidaan esittää valopylväillä. Graafisella käyttöliittymällä varustetussa järjestelmässä on mahdollista esittää esimerkiksi signaalin oskilloskooppikuva tietokoneen näytöllä. Käytettävyyden kannalta olisi kuitenkin hyvä, että järjestelmä osaisi asettaa parametrinsa automaattisesti vastaanotetun signaalin mukaisiksi ja säätää myös vastaanottimen taajuuden tarkasti kohdalleen.

Parametrien automaattisen asettamisen lisäksi tutkitaan mahdollisuuksia kaukokirjoitussignaalien automaattiseen tunnistamiseen ja hakemiseen.

6.1. Vastaanottimen ohjaaminen

Kaukokirjoitussignaalien hakua varten tarvitaan muutamia toimintoja, joilla tietokone voi ohjata vastaanotinta. Tässä toteutuksessa vastaanottimena käytetään Yaesu-Musen FT-920 -lyhytaaltolähetinvastaanotinta, jota on mahdollista ohjata RS-232-sarjaliitintä käyttäen. Kuten useat muutkin saman valmistajan laitteet, tämä lähetinvastaanotin käyttää ns. *CAT-protokollaa* (Computer Aided Transceiver). Samasta protokollasta huolimatta eri laitteiden komennot poikkeavat jossakin määrin toisistaan. Tässä esitellään FT-920-lähetinvastaanottimen käyttämä CAT-toteutus.

6.1.1. CAT-protokolla

CAT-protokollalla radiolaitetta ohjataan tietokoneen sarjaportin kautta lähetettävillä komennoilla. Tämä mahdollistaa monien toimintojen automatisoinnin. Useissa muissa samaa protokollaa käyttävissä laitteissa tarvitaan erillinen RS-232-sovitin, mutta FT-920-lähetinvastaanotimessa on sisäänrakennettu taso-sovitin. Tietokone voidaan kytkeä suoraan laitteeseen tavallisella sarjaliitän-

täkaapelilla, jonka radioon kytkettävä pää on varustettu 9-napaisella D-uroslit-
timellä. Käytettävä yhteysnopeus on 4800 tavua sekunnissa. [Yaesu, 1997]

CAT-protokollassa ohjauksen komennot ovat viiden tavun mittaisia. Komen-
non neljä ensimmäistä tavua ovat parametreja ja viimeinen tavu sisältää varsi-
naisen *operaatiokoodin* (opcode). Kukin tavu koostuu yhdestä aloitusbitistä,
kahdeksasta databitistä ja kahdesta lopetusbitistä. Pariteettibittiiä ei käytetä.
Komennon rakenne on esitetty kuvassa 13.

P4	P3	P2	P1	Operaatiokoodi
----	----	----	----	----------------

Kuva 13. CAT-komennon rakenne

FT-920:n käyttämä protokolla tuntee 19 eri komentoa, mutta kaukokirjoi-
tusjärjestelmässä ei välttämättä tarvita näitä kaikkia. Taulukossa 5 on esitelty
vain kaukokirjoitusjärjestelmässä käytettävät komennot.

Taulukko 5. Käytettävät CAT-komennot

Toiminto	Operaatiokoodi
Taajuuden asettaminen	0x0A
Lähetyslajin asettaminen	0x0C
Taajuuden lukeminen	0x10
Kaistanleveys	0x8C

Sarjaportti on avattava ennen komentojen lähettämistä. Tämä tapahtuu
kohdassa 4.3 esitellyllä tavalla. Nopeudeksi asetetaan 4800 tavua sekunnissa.
Koska komennot sisältävät joitakin perinteisesti tulostuksen ohjaamiseen käy-
tettyjä ASCII-merkkejä, täytyy varmistaa ettei sarjaportti suorita näille mitään
muutoksia, kuten joissakin ympäristöissä tapahtuu. Esimerkiksi taajuuden aset-
tamisessa käytettävä arvo 10 (0x0A) tarkoittaa ASCII-koodissa normaalisti uu-
den rivin aloittamista (new line) tulostuksessa. Oletusasetus saattaa olla se, että
tämän merkin eteen lisätään ASCII-merkki 13 eli vaunun palautus (carriage re-
turn). Tämä aiheuttaa virheellisen toiminnon, koska CAT-komennon operaati-
okoodiksi tuleekin tällöin arvo 13 ja tarkoitettu arvo 10 jää komennon ulko-
puolelle. Tämän vuoksi täytyy varmistaa, ettei sarjaportissa tehdä mitään
muunnoksia. POSIX-standardissa on tälle ominaisuudelle varattu vakio `OPOST`
sarjaportista lähtevää dataa ohjaavalle lipulle `c_oflag`. Tämä vakio asettaa läh-
tevän datan jälkikäsitteilyn päälle ja tässä tapauksessa se täytyy siis asettaa pois

päältä. Jos halutaan estää vain ASCII-merkin 13 lisääminen merkin 10 eteen, mutta sallia muu datan käsittely sarjaportissa, voidaan käyttää nimenomaan tähän tarkoitettua vakiota `ONLCR`, joka täytyy asettaa pois päältä [Sweet, 1999]. Näiden asetusten määrittäminen voi tapahtua esimerkiksi ohjelmakatkelman 10 mukaisesti.

Ohjelmakatkelman 10.

```
settings.c_oflag &= ~OPOST;
settings.c_oflag &= ~ONLCR;
```

CAT-komentojen lähettämistä varten luodaan funktio, joka lähettää parametrina saamansa merkkijonomuotoisen komennon sarjaportille `write`-funktiota käyttäen. Jokaisen lähetettävän tavun väliin lisätään 50 millisekunnin viive. Funktion toteutus on esitetty ohjelmakatkelmassa 11.

Ohjelmakatkelman 11.

```
int yaesu_cmd(int fd, int *command)
{
    int i, succ;

    for(i=0;i<5;i++) {
        delay(50);
        succ=write(fd,(char*) &command[i],1);
        if (succ<0) fputs("Write failed\n",stderr);
    }
    return 0;
}
```

6.1.2. Lähetyslajin ja välitaajuussuodattimien asettaminen

Lähetyslaji asetetaan operaatiokoodilla `0x0C`. Kaukokirjoituksen vastaanotossa käytetään yleensä alemman sivunauhan yksisivunauhalähetettä eli LSB:tä. Joskus kuitenkin tarvitaan myös ylemmän sivunauhan, USB:n, vastaanottomahdollisuutta. FT-920 lähetinvastaanottimessa on myös dataläheteiden vastaanottoon omat asetuksensa, jotka mahdollistavat kapeampien suodattimien käytön. Myös näille lähetyslajeille on omat koodinsa sekä alemmalle että ylemmälle sivunauhalle. Taulukon 6 mukainen lähetyslajia vastaava koodi annetaan komennon viimeisenä parametrina P1, siis juuri ennen operaatiokoodia. Käyttämättömien parametrien P2, P3 ja P4 arvoiksi asetetaan `0xFF`.

Taulukko 6. Parametrin P1 arvot tarvittaville lähetyslajeille.

Lähetyslaji	Parametri P1
LSB	0x00
USB	0x01
DATA-L	0x08
DATA-U	0x0A

Välitaajuussuodattimien leveys asetetaan omalla komennollaan, jonka operaatiokoodi on 0x8C. Leveämpi suodatin otetaan käyttöön parametrin P1 arvolla 0x00, kapeampi suodatin taas arvolla 0x02. Parametrien arvot määritellään vakioina.

Koska lähetyslajin sekä kaistanleveyden valitseminen liittyvät läheisesti toisiinsa, toteutetaan näiden asettaminen samassa funktiossa, joka on esitetty ohjelmakatkelmassa 12.

Ohjelmakatkelman 12.

```
int set_mode(int fd, int mode, int filter)
{
    int cmd[5];

    cmd[0] = 0xFF;
    cmd[1] = 0xFF;
    cmd[2] = 0xFF;
    cmd[3] = mode;
    cmd[4] = 0x0C;
    yaesu_cmd(fd,cmd);

    cmd[3] = filter;
    cmd[4] = 0x8C;
    yaesu_cmd(fd,cmd);

    return 0;
}
```

Jos halutaan asettaa lähetyslajiksi LSB-datalähetteen vastaanotto kapeampaa välitaajuussuodatinta käyttäen, on funktion kutsu ohjelmakatkelman 13 mukainen.

Ohjelmakatkelman 13.

```
#define DATAL 0x08
#define NARROW 0x02

set_mode(fd, DATAL, NARROW);
```

6.1.3. Taajuuden asettaminen

Vastaanotettava taajuus asetetaan operaatiokoodilla 0x0A. Asetettava taajuus annetaan binäärikoodatussa desimaalimuodossa ja käänteisessä järjestyksessä siten, että viimeiseksi annetaan operaatiokoodi. Taulukossa 7 on esitetty parametrit erään taajuuden asettamiseksi.

Taulukko 7. Parametrien arvot taajuuden 14.256,78 MHz asettamiseksi

Parametri	Arvo
P4	0x78
P3	0x56
P2	0x42
P1	0x01
operaatiokoodi	0x0A

Taajuuden asettamista varten toteutetaan ohjelmakatkelmassa 14 esitetty funktio, joka saa parametrinaan taajuuden kilohertsin sadasosina, siis kymmeninä hertseinä.

Ohjelmakatkelmä 14.

```
int set_freq(int fd, int freq)
{
    int i, div, cmd[5];

    cmd[4] = 0x0A;
    div = 10000000;
    for (i=3;i>=0;i--); {
        cmd[i] = 16*(freq/div)+(freq%div)/(div/10);
        freq = freq%(div/10);
        div = div/100;
    }
    yaesu_cmd(fd,cmd);
    return 0;
}
```

Mikäli halutaan asettaa käytettäväksi taajuudeksi 10.102,40 kHz, funktiota kutsutaan näin: `set_freq(fd, 1010240);`

6.2. Vastaanottoparametrien automaattinen asettaminen

Parametrien asettamista varten vastaanotettavasta äänisignaalista täytyy tunnistaa kaukokirjoitussignaalin käyttämä äänitaajuuspari. Saadun taajuustiedon perusteella voidaan vastaanottimen kuuntelemaa taajuutta virittää siten, että äänitaajuuspari osuu parametrien mukaisille suodattimien päästökaistoille. Vaihtoehtoisesti järjestelmä voi asettaa kyseiset parametrit vastaanotettavalle signaalille sopiviksi. Digitaalista signaalinkäsittelyä hyödyntävä järjestelmä si-

sältää valmiiksi käyttökelpoista tietoa automaattisen virityksen toteuttamiseen. Analysoiminen tapahtuu muodostamalla signaalista profiili, josta selviää äänisignaalin sisältämät eritaajuiset komponentit, siis signaalin amplitudi ääntaajuuksien funktiona. Sopiva työkalu tähän tehtävään on *nopea Fourier-muunnos* (FFT, Fast Fourier Transform) [Lynn and Fuerst, 1998].

6.2.1. Fourier-muunnoksista

Fourier-muunnosta käytetään signaalin sisältämien taajuuksien analysointiin siten, että signaali hajotetaan sisältämiinsä komponentteihin. Fourier-muunnoksesta on erilaisia variaatioita eri tyyppisten signaalien analysointiin. Signaalit voidaan jakaa jatkuva-aikaisiin ja diskreettiaikaisiin sekä toisaalta jaksollisiin ja jaksottomiin signaaleihin. Kaukokirjoitusjärjestelmässä äänikortilta saatavat signaalinäytteet muodostavat jaksollisen diskreettiaikaisen signaalin. Tällaisen analysointiin sopiva algoritmi Fourier-muunnosten perheessä on *diskreetti Fourier-muunnos* (DFT, Discrete Fourier Transform) [Smith, 1999].

Jaksollinen diskreettiaikainen signaali voidaan hajottaa sinimuotoisiin komponentteihin, joista kunkin taajuus on vakio, mutta niiden amplitudi voi vaihdella. Jos signaali koostuu N :stä kappaleesta näytteitä, se voidaan hajottaa $N/2+1$:een sinikomponenttiin ja $N/2+1$:een kosinikomponenttiin [Smith, 1999].

Diskreetti Fourier-muunnos voidaan toteuttaa tutkimalla annetun signaalin ja vertailusignaalin välistä korrelaatiota. Jokaista signaalinäytettä verrataan referenssisignaaliin ja selvitetään, onko näytteessä referenssisignaaliin sopivia komponentteja. Tämä voidaan tehdä seuraavia *analyysiyhtälöitä* (analysis equation) käyttäen:

$$X[k]_{re} = \sum_{i=0}^{N-1} x[i] \cos\left(\frac{2\pi ki}{N}\right) \quad (6.2.1)$$

ja

$$X[k]_{im} = -\sum_{i=0}^{N-1} x[i] \sin\left(\frac{2\pi ki}{N}\right). \quad (6.2.2)$$

Ensimmäisen kaavan (6.2.1) mukaisella yhtälöllä jokainen aikatason signaalinäytteen x piste $x[i]$ kerrotaan kosiniaallon vastaavalla pisteellä. Kerroin k määrittelee verrattavan kosiniaallon taajuuden. Summaamalla nämä tulot kaikkien näytteen pisteiden yli saadaan tuloksena reaalitasen arvo $X[k]_{re}$ kerrointa k vastaavalle taajuudelle. Vastaavasti toisen kaavan (6.2.2) mukaisella yhtälöllä saadaan imaginääritason arvo $X[k]_{im}$. Toistamalla tämä kullekin haluttua taa-

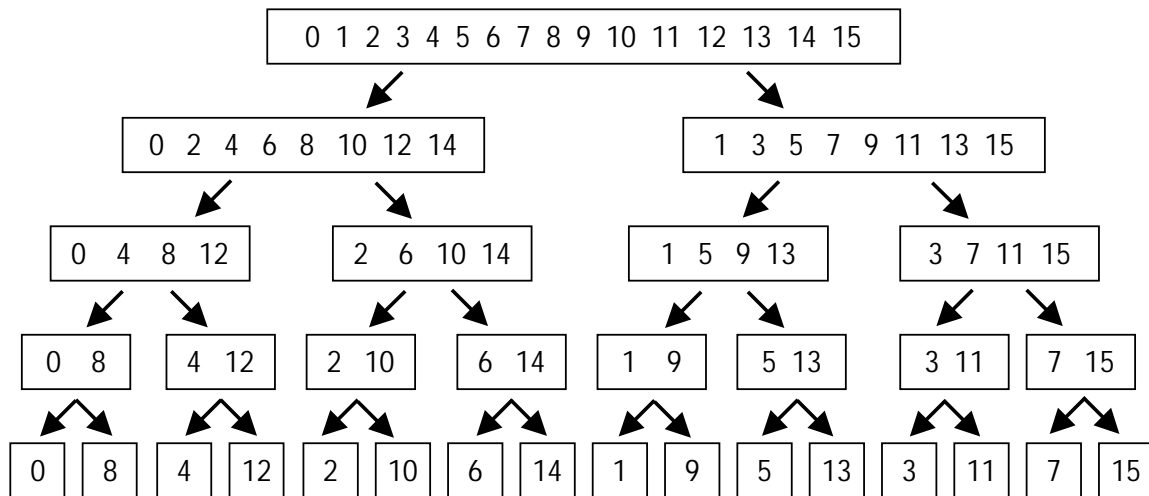
juuskaistaa vastaavalle kertoimen k arvolle, saadaan muodostettua signaalin taajuusspektri [Smith, 1999].

Nopealla Fourier-muunnoksella tarkoitetaan joukkoa tehokkaita algoritmeja, joilla voidaan pienentää diskreetin Fourier-muunnoksen aikavaativuutta. Näitä voidaan soveltaa erityisesti tehokkuutta vaativiin digitaalisen signaalinkäsittelyn sovelluksiin, esimerkiksi signaalin spektrin analysointiin [Lynn and Fuerst, 1998]. Nopea Fourier-muunnos soveltuu periodiselle jonolle, jonka jakso on jokin kakkosen potenssi, siis $N=2^k$. Tällöin pystytään laskemaan Fourier-muunnos jonolle N laskemalla muunnokset kahdelle jonolle, joiden pituudet ovat $N/2$. Sama voidaan toistaa edelleen myös näille jonoille ja näin jatketaan kunnes käsiteltävänä on yhden näytteen mittainen jono.

Nopean Fourier-muunnoksen algoritmi on seuraava [Smith, 1999]:

1. Mikäli muunnettavan lukujonon $x(n)$ dimensio on 1, palautetaan arvo sellaisenaan. Kyseisen signaalin spektri on siis $X(n) = x(n)$.
2. Jaetaan lukujono $x(n)$ kahdeksi osalukujonoksi siten että jono $x_0(n)$ sisältää ne näytteet, joiden indeksi on parillinen ja jono $x_1(n)$ taas näytteet, joiden indeksi on pariton.
3. Lasketaan lukujonojen $x_0(n)$ ja $x_1(n)$ Fourier-muunnokset rekursiivisesti tällä algoritmilla.
4. Yhdistetään vektorit pareittain, kunnes tuloksena on yksi taajuustason spektriä vastaava vektori. Yhdistäminen tapahtuu käänteisessä järjestyksessä vaiheessa 2 suoritettuun purkuun nähden.

Nopea Fourier-muunnos perustuu siis ongelman hajottamiseen pienempiin osaongelmiin. Signaalinäytteistä koostuvan lukujonon jakaminen osajonoihinsa tehdään *lomittavaa purkua* (interlaced decomposition) käyttäen, kuvan 14 esittämällä tavalla.



Kuva 14. Lomittava purku

Tietokoneella lomitusta voidaan toteuttaa binäärimuotoon muutettujen indeksien kääntämisellä (bit reversal). Tämä tapahtuu muuntamalla jonon indeksit ensin binäärimuotoon, jonka jälkeen kyseiset binääriluvut muunnetaan peilikuvikseen. Tämän jälkeen jono järjestetään saatujen uusien binääri-indeksien mukaiseen järjestykseen. Taulukko 8 esittää uusien indeksien muodostamisen 8-alkioiselle jonolle tällä menetelmällä [Smith, 1999].

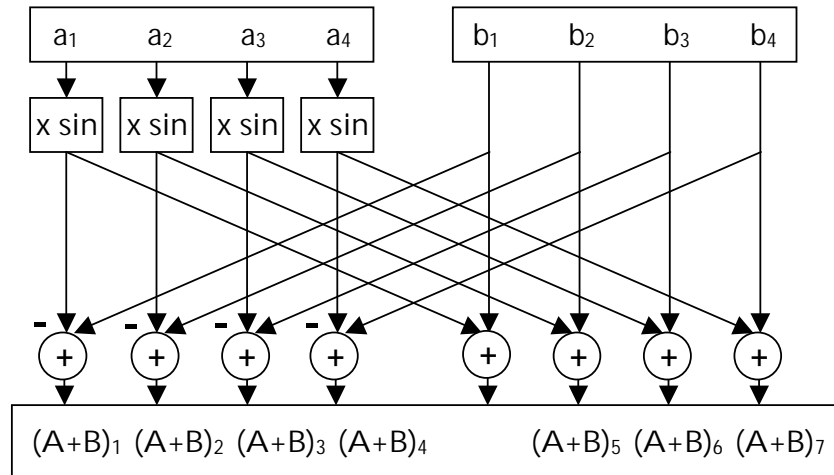
Taulukko 8. Esimerkki lomituksen toteuttamisesta binäärilukujen kääntämistä käyttäen

Alkuperäiset indeksit			Uudet indeksit	
Desim.	Bin.		Bin.	Desim.
0	000	→	000	0
1	001	→	100	4
2	010	→	010	2
3	011	→	110	6
4	100	→	001	1
5	101	→	101	5
6	110	→	011	3
7	111	→	111	7

Rekursiosta ansiosta varsinainen muutos aikatasolta taajuustasolle on trivi-aali, koska yksikkösignaalin taajuusspektri sisältää vain itsensä. Saatujen osavektoreiden yhdistäminen koko taajuusspektriä on algoritmin hankalin vaihe

[Smith, 1999]. Mikäli hajotus on tehty kuvan 14 mukaisesti, yhdistettävänä on 16 yhden pisteen kokoista taajuusspektriä.

Ensimmäisessä vaiheessa nämä yhdistetään pareittain, jolloin tuloksena on 8 kahden pisteen kokoista taajuusspektriä. Tätä jatketaan, kunnes on saatu koostettua yksi vektori, joka muodostuu kuvan 14 tapauksessa 16 pisteestä. Vektori esittää siis näytteistetyn signaalin taajuusspektrin. Yhdistettävien taajuusspektrivektoreiden pisteet koostuvat kompleksiluvuista.



Kuva 15. Taajuusspektrivektoreiden uudelleen kokoaminen

Kompleksiluvuista koostuvan kahden taajuustason vektorin yhdistäminen tapahtuu siten, että ennen yhdistämistä toinen vektoreista kerrotaan sinimuotoista signaalia kuvaavalla vektorilla. Tämän jälkeen vektorit yhdistetään pisteittäin siten, että kunkin kahden pisteen tuloksena syntyy kaksi uuden vektorin pistettä, toinen summaamalla ja toinen erotuksena kuvan 15 esittämällä tavalla [Smith, 1999]. Yhdistämisen jälkeen tuloksena on vektori, joka kuvaa annetun signaalinäytteen spektrin taajuustasossa.

Nopean Fourier-muunnoksen toteuttavia algoritmeja on saatavana valmiina kirjastoina esimerkiksi c-kielellä toteutettavissa ohjelmissa käytettäväksi [Press *et al.*, 1997].

6.2.2. Ikkunoinnista

Fourier-muunnoksen tuottamassa taajuustason spektrissä esiintyy värähtelyä yksittäisen spektriipiikin ympäristössä. Osa piikin energiasta *vuotaa* (spectral leakage) sitä ympäröiville taajuuksille ja spektrissä näkyy siis piikkejä taajuuksilla, joilla ei signaalia todellisuudessa ole. Tämän ongelman eliminointiin voidaan käyttää erilaisia *ikkunointimenetelmiä* (windowing). Näissä aikataso sig-

naalinäytteiden muodostama vektori kerrotaan sopivaa käyrää kuvaavalla vektorilla. Lopputuloksena saadaan signaalivektori, joka on käyrän muodon mukaisesti painottunut. Kun ikkunoidulle signaalille suoritetaan Fourier-muunnos, on saatava spektri hieman tasaisempi ja todelliset huiput saadaan paremmin esiin. Ikkunointikäyrien määrittelyyn käytetään erilaisia ikkunointifunktioita. Tunnettuja funktioita ovat mm. Hamming-, Hanning-, Bartlett- ja Blackman-funktiot sekä suorakulmainen ikkunointifunktio. Funktiot poikkeavat toisistaan esimerkiksi kaistanleveydeltään ja värähtelyominaisuuksiltaan. Ne soveltuvat siten erilaisiin tehtäviin suodattimien optimoinnissa ja signaalin analysoinnissa [Lynn and Fuerst, 1998]. Kaukokirjoitusjärjestelmässä signaalin spektrin analysointia optimoidaan Hanning-funktiolla, joka on yksinkertainen, mutta ominaisuuksiltaan riittävä. Hanning-funktio tunnetaan kehittäjänsä mukaan myös *van Hannin* funktion nimellä [Lynn and Fuerst, 1998]. Funktion tuottaman käyrän muodosta johtuen funktiosta puhutaan myös *nostettuna kosini-funktiona* (raised cosine function) [Smith, 1999]. Kirjallisuudessa tavallisimmin esiintyvä nimitys kuitenkin on Hanning-funktio, vaikka nimi onkin tavallaan virheellinen. Hanning-funktio on muotoa

$$w[i] = 0.5 + 0.5 \cos\left(\frac{2\pi i}{N}\right). \quad (6.2.3)$$

Kaavan (6.2.3) tuottama käyrä vastaa pituudeltaan signaalinäytteiden yhden jakson pituutta N . Funktion toteutus c-kielisenä on esitetty ohjelmakatkelmassa 15. Jokainen signaalinäytevektorin näytteen reaali- ja imaginääriosa kerrotaan erikseen kaavan vastaavan indeksin tuottamalla arvolla. Tuloksena saadaan uusi kompleksilukuvektori, joka voidaan antaa edelleen FFT-algoritmin käsiteltäväksi.

Ohjelmakatkelmä 15.

```
void hanning(complex *x)
{
    static float w[NSAMPLES];
    static int first=1;
    int i;

    if (first) {
        first=0;
        for (i=0;i<NSAMPLES;i++) w[i]=(1-cos(i*2*M_PI/NSAMPLES))*0.5;
    }
    for (i=0;i<NSAMPLES;i++) {
        x[i].re*=w[i];
        x[i].im*=w[i];
    }
}
```

6.2.3. MARK- ja SPACE-signaalien hakeminen

Alakohdissa 6.2.1 ja 6.2.2 kuvatuilla menetelmillä saadaan suoritettua signaalinäytteen spektrianalyysi. Tuloksena on signaalin taajuustason profiili eli kuvaus signaalin sisältämistä äänitaajuuksista sekä kutakin taajuutta edustavien komponenttien amplitudeista. Yhden jakson mittainen näyte antaa tietoa hetkellisestä tilanteesta.

Ohjelmakatkelman 16 mukainen funktio hakee yhden jakson mittaisesta näytteestä taajuuden, jolla suurin amplitudihuippu esiintyy. Huipun täytyy ylittää parametrina annettava kynnyisarvo.

Ohjelmakatkelman 16.

```
int peak_freq(int threshold)
{
    unsigned int i,ix,np, xstep, maxx, maxy;
    static xy spectrum[NSAMPLES];
    complex a[NSAMPLES];

    ix=ctrlrdp->inp+1;
    for (i=0; i<NSAMPLES; i++) {
        ix&=CTRLBUF-1;
        a[i].im=0.0;
        a[i].re=ctrlrdp->in[ix++]/128.0;
    }
    hanning(a);          /* ikkunointi Hanning-funktiolla */
    fft(a);              /* nopea Fourier-muunnos */
    db(a);               /* amplitudin erottaminen */
    np=NSAMPLES;
    xstep=FS/NSAMPLES; /* askel=naytteenottotaaj./jakson pituus */
    for (i=0;i<np;i++) {
        spectrum[i].x=i*xstep;
        spectrum[i].y=a[i].re*10;
    }
    maxx=0;maxy=threshold; /* Kynnyisarvo */
    for (i=10;i<np;i++) {
        if ( spectrum[i].y>maxy ) {
            maxy=spectrum[i].y;
            maxx=spectrum[i].x;
        }
    }
    return(maxx);
}
```

Koska kaukokirjoituksessa MARK- ja SPACE-taajuiset signaalit vuorottelevat, toistetaan ohjelmakatkelman 16 mukaisen funktion kutsua molempien taajuuksien löytämiseksi. Tätä jatketaan, kunnes peräkkäiset kutsut palauttavat kaksi eri taajuutta, jotka poikkeavat toisistaan riittävästi. Näin löydettyjen MARK- ja SPACE-taajuuksien perusteella lasketaan suodattimille uudet kertoimet.

Ohjelmakatkelmassa 17 esitetty funktio toteuttaa tämän toiminnon. Funktiota voidaan esimerkiksi kutsua näppäimistökomennolla, mikäli halutaan asettaa parametrit vastaanottimesta kuuluvalla signaalille sopiviksi. Automatisoidussa järjestelmässä ohjelma voi kutsua funktiota säännöllisin väliajoin ja pitää siten parametrit sopivina.

Ohjelmakatkkelma 17.

```
void snapshot(int diff, int tsh1, int tsh2)
{
    int pk1=0,pk2=0;

    while ((pk1==pk2) || (pk2-pk1<diff)) {
        pk1=peak_freq(tsh1);
        pk2=peak_freq(tsh2);
    }

    radiodp->lock=1;
    radiodp->peak1=pk1;
    radiodp->peak2=pk2;

    fskdp->spacef=pk1;
    fskdp->markf=pk2;
    set_filters(fskdp->spacef,fskdp->markf,fskdp->dr);
}
```

6.3. Kaukokirjoituslähetyksen tunnistaminen

Lähetyksen tunnistamista voidaan käyttää kaukokirjoitussignaalin hakemiseen tietyltä taajuusalueelta. Järjestelmä käy läpi ennalta määrättyä taajuusaluetta ja tunnistaa siellä esiintyviä signaaleita sekä ilmoittaa, mikäli taajuusalueelta löytyy kaukokirjoitussignaali. Tällaista menetelmää voidaan käyttää silloin, kun vastaanotettavan kaukokirjoitusaseman tarkka taajuus ei ole tiedossa. Tätä voidaan hyödyntää esimerkiksi radiotiedustelussa.

Hakujärjestelmä voi toimia esimerkiksi siten, että se käy läpi tiettyä lyhytaaltotaajuusaluetta ja löydettyään hakuparametrien mukaisen signaalin, vastaanottaa sitä muutaman minuutin ajan tallentaen vastaanottamansa tekstin tiedostoon taajuustiedoilla ja aikaleimalla varustettuna.

6.3.1. Lähetyksen tunnistaminen signaalin perusteella

Kohdassa 6.2 esiteltyä menetelmää voidaan soveltaa myös lähetteen tunnistamiseen. Mikäli taajuustasoon muodostetussa signaaliprofiilissa esiintyy kaksi suunnilleen samantasoista amplitudihuippua, saattaa kyseessä olla kaukokirjoitussignaali. Lisäksi voidaan tarkastella amplitudihuippujen etäisyyttä toisistaan eli sitä, vastaako etäisyys kaukokirjoituksessa käytettyjä taajuussiirtymiä.

Kaukokirjoitussignaalin hakemiseksi tietyltä vastaanottimen taajuusväliltä, luodaan sääntöjä, joiden perusteella hakualgoritmi ohjaa vastaanotinta. Aluksi määritellään hakuparametrit eli haettavan kaukokirjoitinlähetteen AFSK-taajuuspari ja symbolinopeus sekä taajuusväli, jolta signaalia haetaan.

Vastaanottimessa lähetyksilajiksi asetetaan LSB eli vaimennetun kantoaallon alapuolinen yksisivunauhalähete. Tällöin vastaanotettavien äänisignaalien taajuus muuttuu samansuuntaisesti vastaanottimen taajuuden muutoksen kanssa,

eli siirryttäessä vastaanottimen taajuusasteikolla alaspäin, myös kuultavan äänisignaalin taajuus laskee.

Haku aloitetaan määritellyn taajuusvälin ylärajalta ja kuunnellaan asetettua taajuutta sekä muodostetaan äänitaajuusprofiili. Mikäli profiili ei sisällä haku-parametreihin sopivia äänitaajuuksia, viritetään vastaanottimen taajuus yhden kilohertsin verran nykyisen taajuuden alapuolelle, ja muodostetaan uusi äänitaajuusprofiili.

Mikäli profiilista löytyy kaksi suunnilleen samantasoista amplitudihuippua, tarkastetaan niiden välinen etäisyys taajuustasossa. Jos etäisyys vastaa haettavan signaalin AFSK-taajuusparia, viritetään vastaanottimen taajuutta siten, että oletetun signaalin amplitudihuiput osuvat kaukokirjoitusjärjestelmän suodatimiin. Tämän jälkeen signaalin tarkastelua voidaan jatkaa sisällön perusteella.

6.3.2. Lähetyksen tunnistaminen sisällön perusteella

Vaikka signaali olisi tunnistettu taajuuspektrinsä perusteella kaukokirjoituslähetteeksi, saattaa kyseessä kuitenkin olla esimerkiksi kaksi muun tyyppistä erillistä lähetettä, jotka sattuvat olemaan taajuustasolla sopivalla etäisyydellä toisistaan. Tunnistamista voidaan varmentaa tarkastelemalla myöskin vastaanotetun lähetyksen sisältöä. Ensimmäinen vaihe sisällön tarkastelussa on vastaanotetun datan toteaminen Baudot-koodatuksi.

Vastaanotetusta koodista tarkastetaan bittijonon rakenne dekodausvaiheessa ennen merkkien muodostamista. Tämä tarkoittaa sitä, että kukin yhtä merkkiä vastaava bittijono alkaa SPACE-tasoisella aloitusbitillä, ja että viiden varsinaisen databitin jälkeen seuraa lopetusbitti, joka normaalin Baudot-koodin tapauksessa on pituudeltaan puolitoistakertainen muihin bitteihin nähden. Myös pariteettitarkastus tehdään, mikäli pariteettibitti on käytössä vastaanotetavassa koodissa. Häiriöistä johtuen merkkivirheitä ja pariteettivirheitä tietysti jonkin verran esiintyy, mutta mikäli vastaanotettu koodi pääsääntöisesti läpäisee kummatkin tarkastukset, voidaan signaali olettaa Baudot-koodatuksi.

Sisältöön perustuvaa tarkastelua voidaan edelleen täsmentää, mikäli tiedossa on jokin vastaanotettavan informaation ominaisuus, esimerkiksi käytetty kieli, sisältöä koskeva muutosääntö tai todennäköinen merkkijono. Monet kaukokirjoituslähetykset koostuvat esimerkiksi määrätyn mittaisista numerosarjoista, joten tätäkin tietoa voidaan käyttää tunnistetietona automaattisessa kaukokirjoituslähetyksen hakemisessa ja tunnistamisessa. Esimerkiksi merisääinformaatiota lähetetään numerosarjoista koostuvaa SYNOP FM 12 -koodia käyttäen.

7. Toteutetusta järjestelmästä

7.1. Menetelmien vertailua

Ulkoiseen demodulaattoriin perustuva järjestelmä, joka esiteltiin luvussa 4, jäi lähinnä vertailukohdaksi DSP-pohjaista järjestelmää toteutettaessa. Menetelmiä vertailtiin kokeilemalla kumpaakin erilaisten kaukokirjoitussignaalien vastaanottoon. Vaikka varsinaisia suorituskykymittauksia ei suoritettu, menetelmien kokeilu erilaisissa häiriöolosuhteissa antoi selkeän kuvan kummankin menetelmän suorituskyvystä. Digitaaliseen signaalinkäsittelyyn pohjautuvan järjestelmän edut ovat varsin ilmeiset. Koska suodattimien arvoja, samoin kuin muitakin parametreja, voidaan muuttaa tarpeen mukaan, on DSP-pohjainen järjestelmä erittäin joustava.

Modeemiin toteutettuna erilaisille signaaleille sopiva demodulaattori on suorituskyvyltään varsin heikko, koska suodattimet ovat tällöin päästökaisaltaan leveitä ja vastaanotettavaan signaaliin sekoittuu häiriöitä ja muita ei-haluttuja signaaleja. Jos taas haluttaisiin toteuttaa hyvin kapeakaistainen demodulaattori perinteisellä menetelmällä, siitä tulisi kyseisillä komponenttiarvoilla vain tietyn tyyppiselle signaalille sopiva. Tällainen järjestelmä ei sovellu myöskään kovin hyvin vastaanoton automatisointiin, koska se ei tarjoa juurikaan tietoa, jota voitaisiin hyödyntää vastaanottimen ohjaamisessa.

Luvussa 5 esitelty DSP-pohjainen järjestelmä osoittautui joustavuutensa lisäksi häiriönsietokyvyltäänkin varsin onnistuneeksi. Signaalinäytteitä analysoimalla myös vastaanottoparametrien automaattinen asettaminen onnistuu tässä järjestelmässä varsin helposti. Näin ollen tässä tutkielmassa esitetyt menetelmät radiokaukokirjoituksen vastaanoton automatisoimiseksi pohjautuvat digitaalista signaalinkäsittelyä hyödyntävällä järjestelmällä tehtyihin kokeiluihin.

7.2. Järjestelmän arkkitehtuuri

Ohjelma koostuu useammasta säikeestä, joiden välinen kommunikaatio tapahtuu kaikille säikeille näkyvän jaetun muistialueen kautta. Säikeet on toteutettu POSIX-säikeinä pthreads-kirjastoa käyttäen.

Kun ohjelma käynnistetään, pääohjelma lukee aluksi vastaanottoparametrit konfiguraatitiedostosta. Mikäli konfiguraatitiedosto puuttuu, käytetään oletusparametreja. Ohjelma laskee parametrien mukaisesti esimerkiksi demodulaattorin suodattimien kertoimet.

Pääohjelma käynnistää kaksi säiettä, joista toinen tarkkailee käyttäjän antamia näppäinkomentoja ja näiden perusteella esimerkiksi askeltaa vastaanotettavan koodin nopeutta tai käytettäviä AFSK-taajuuspareja.

Toinen säie toteuttaa järjestelmän automaatiotoiminnot. Mikäli automaattinen virittäminen on aktivoitu, säie tarkkailee vastaanotettavaa äänisignaalia ja asettaa demodulaattorin suodatinkertoimet signaalille sopiviksi.

Itse pääohjelma siirtyy silmukkaan, joka suorittaa signaalin demoduloinnin, dekodauksen ja merkkien tulostamisen näytölle eri funktioita kutsumalla.

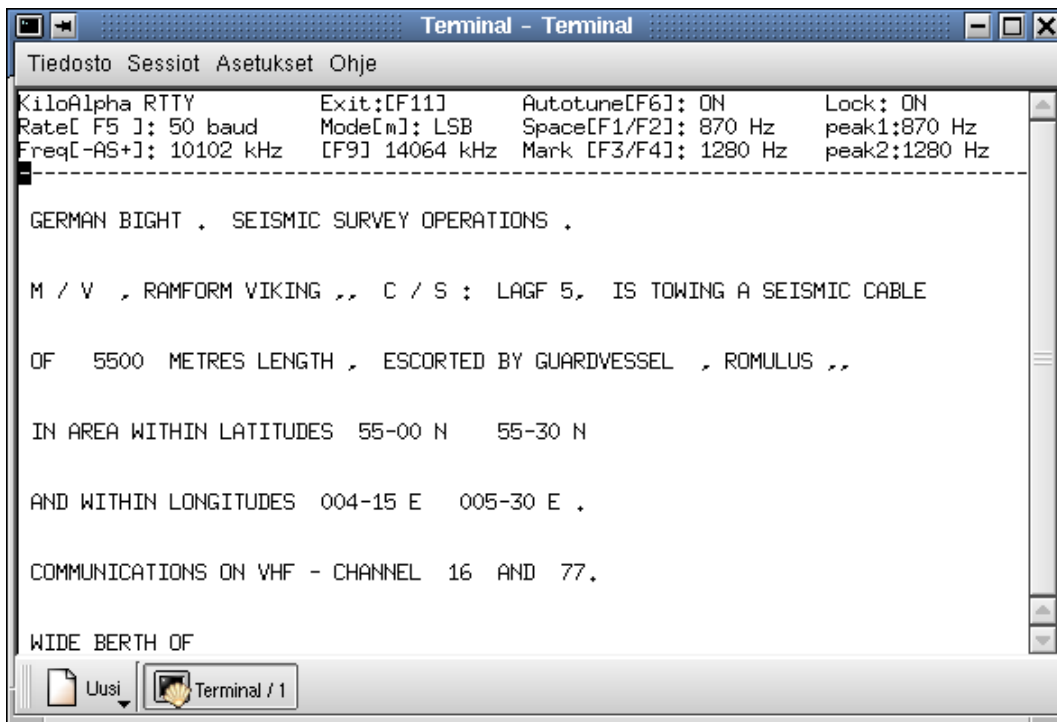
Mikäli hallintasäie vastaanottaa käyttäjältä lopetuskäskyn, se välittää tästä tiedon jaetun muistialueen kautta muille säikeille, jotka lopettavat itsensä. Tällöin myös dekodaussilmukan suorittaminen pysähtyy.

7.3. Käyttöliittymä

Järjestelmälle asetettujen vaatimusten mukaisesti toteutettiin käyttöliittymä, jota on mahdollista käyttää merkkipohjaisella päätteellä tai pääteohjelmalla.

Näyttö on jaettu väliiviivalla kahteen osaan. Näytön kolme ylimmäistä riviä muodostavat varsinaisen käyttöliittymän, joka kertoo järjestelmän voimassaolevat vastaanottoparametrit sekä näppäinkomennot, joilla näitä voidaan säätää. Esimerkiksi F6-funktionäppäimellä voidaan asettaa vastaanottoparametrien automaattinen asettaminen päälle tai pois. Vasemmassa ylänurkassa oleva Lock-kenttä kertoo, mikäli automaattiasetus on lukinnut parametrit vastaanotettavan signaalin mukaisiksi.

Vastaanotettu kaukokirjoitusteksti rullaa näytön alaosan muodostamassa vastaanottoikkunassa. Käyttöliittymän ikkunointi on toteutettu ncurses-kirjasto [Goldt *et al.*, 1995] käyttäen. Kuva 16 esittää näkymän käyttöliittymään X-ikkunointiympäristön pääteohjelmassa.



Kuva 16. Toteutettu käyttöliittymä

7.4. Jatkokehitysmahdollisuuksia

Tämän tutkielman yhteydessä toteutettu järjestelmä sisältää vain perusominaisuudet radiokaukokirjoituksen vastaanottoon sekä menetelmän vastaanotto-parametrien automaattiseen asettamiseen. Työn aikana on kuitenkin syntynyt ideoita järjestelmän jatkokehitystä varten.

Koska DSP-pohjaisen järjestelmän ominaisuudet ovat helposti säädettävissä erilaisille vaihtotaajuusavainnuksen taajuuspareille ja nopeuksille, olisi mahdollista kehittää järjestelmää myös muiden vaihtotaajuusavainnusta käyttävien menetelmien vastaanottoon. Tähän tarvittaisiin dekodausalgoritmeja Baudotkoodin lisäksi myös muille koodauksille. Kohdassa 2.4 esiteltyjen virheenkorjaavien menetelmien vastaanottaminen vaatisi dekodauksen lisäksi myös ylempien protokollatasojen toteuttamista. Digitaalista signaalinkäsittelyä käyttäen on mahdollista toteuttaa demodulaattori myöskin vaiheensiirtoavainnusta käyttäville menetelmille. Useita tällaisia ohjelmia onkin jo saatavana.

Järjestelmää voisi kehittää jatkojalostamaan vastaanottamaansa informaatiota. Monet kaukokirjoituspalvelut lähettävät informaatiota numerosarjoiksi koodattuna. Esimerkiksi merisäätiedotuksia lähetetään ns. SYNOP FM 12 -koodia käyttäen. Tässä koodissa sääasemien paikkatiedot ja varsinainen sääinformaatio on koodattu viiden numeron sarjoihin, jotka lähetetään tavallista

Baudot-koodattua kaukokirjoitusta käyttäen [Seewetter, 2002]. Järjestelmään voisi siis lisätä toiminnon, joka tulkkaisi vastaanotetun numerokoodin luettavaan muotoon.

Vastaanottotoiminto voisi myös toimia ajastetusti. Useat kaukokirjoituspalvelut, kuten edellä mainittu merisääpalvelu, lähettävät tietyn tyyppistä informaatiota määrättyyn kellonaikaan. Järjestelmä voisi asettaa ajastetusti vastaanottimen taajuuden ja muut parametrit sopiviksi, vastaanottaa halutun informaation ja tallentaa sen tiedostoon.

Useat olemassa olevat radiokaukokirjoitusjärjestelmät käyttävät graafista käyttöliittymää, jolloin on mahdollista esittää näytöllä mm. oskilloskooppikuva signaalin käsittelyn eri vaiheista. Esimerkiksi Jesús Ariasin RTTY-ohjelma sisältää tällaisia toimintoja [Arias, 2001]. Linux-käyttöjärjestelmässä on erilaisia työkaluja graafisen käyttöliittymän toteuttamiseen, lähinnä X-ikkunointiin perustuvissa ympäristöissä. Tällaisia työkaluja ovat esimerkiksi XForms- ja Qt-kirjastot.

Radion ohjaukseen tarvittavat toiminnot olisi mahdollista toteuttaa standardoitua rajapintaa käyttäen, jolloin tuki erimerkkisille ja -tyyppisille vastaanottimille olisi helposti lisättävissä. Tällaisen rajapinnan määrittelytyö on jo käynnissä, mutta toistaiseksi standardi on vielä varsin puutteellinen.

Toteutettu järjestelmä keskittyy pelkästään kaukokirjoituksen vastaanottoon. Mikäli järjestelmää haluttaisiin käyttää kaksisuuntaiseen liikennöintiin, siihen olisi varsin yksinkertaista lisätä myös lähetettävän kaukokirjoitussignaalin muodostaminen äänikortilla.

Koska lyhyillä aalloilla kaukokirjoitussignaalien vastaanottoon tarvittavat antennit ovat varsin kookkaita, on tarpeeksi tehokkaan antennin pystyttäminen kaupunkiolosuhteissa usein ongelmallista. Samoin kaupunkiympäristössä esiintyy usein radiotaajuuksilla esimerkiksi sähkölaitteista johtuvia häiriöitä, jotka vaikeuttavat vastaanottoa. Ongelma voitaisiin ratkaista sijoittamalla antenni vapaaseen tilaan kauas häiriölähteistä, esimerkiksi taajama-alueen ulkopuolelle. Tällöin vastaanottojärjestelmässä pitäisi olla myös kaukokäyttömahdollisuus. Tämä voitaisiin toteuttaa esimerkiksi siten, että järjestelmä toimittaisi dekodatun tekstin TCP/IP-yhteyttä käyttäen toisessa paikassa sijaitsevaan esitysohjelmaan, joka sisältäisi myös järjestelmän ohjaamiseen tarvittavat toiminnot.

8. Yhteenveto

Tämän tutkielman tavoitteena oli perehtyä radiokaukokirjoituksen vastaanotomenetelmiin sekä tutkia mahdollisuuksia vastaanoton automatisointiin.

Vertailtaessa kahta erilaista Baudot-koodatulle kaukokirjoitukselle tarkoitettua vastaanotomenetelmää, muotoutui käsitys digitaalisen signaalinkäsittelyn tarjoamista mahdollisuuksista ja ilmeisistä eduista perinteiseen ulkoisilla komponenteilla toteutettuun demodulaattoriin nähden. DSP-pohjainen järjestelmä on huomattavasti joustavampi ja helpommin muunneltavissa erilaisten signaalien vastaanottamista varten, toisin kuin ulkoinen demodulaattori, joka on yleensä tarkoitettu vain tietyn tyyppisen signaalin vastaanottoon. Käytännön kokeissa DSP:llä toteutettujen suodattimien suorituskyky osoittautui vähintäänkin yhtä hyväksi kuin kokeillun modeemipiirin suodattimien suorituskyky. Tosin hyvälaatuisilla komponenteilla toteutettuna ja tarkkoja mittalaitteita virityksessä käyttäen voidaan ulkoisesta demodulaattorista saada tämän tutkielman yhteydessä toteutettua laitetta huomattavasti häiriöisempi.

Vastaanoton automaatiomenetelmien kokeilussa keskityttiin käyttämään ohjelmallista DSP-toteutusta. Nopea Fourier-muunnos tarjoaa tehokkaan tavan analysoida äänisignaalin sisältämiä taajuuksia. Analysointia voidaan hyödyntää kaukokirjoituksen vastaanotossa varsinkin vastaanottoparametrien automaattisessa asettamisessa. Tästä saadut kokemukset olivat hyvin positiivisia. Käytännössä tätä kokeiltiin virittämällä vastaanotin hieman sivuun kaukokirjoitinsignaalista, jolloin signaali kuuluu vastaanottimesta, mutta signaalin dekodaus ei onnistu. Kun parametrien automaattinen asettaminen aktivoitiin, järjestelmä viritti suodattimet nopeasti kohdalleen ja dekodaus käynnistyi.

Samaa menetelmää voidaan käyttää myös kaukokirjoitussignaalien automaattisessa hakemisessa ja tunnistamisessa. Pelkkä signaalin taajuusprofiiliin perustuva tunnistus helposti tulkitsee erillisiä lähetteitä yhdeksi kaukokirjoitussignaaliksi erityisesti ruuhkaisilla taajuusalueilla. Tämän vuoksi on syytä käyttää tunnistamiseen myös vastaanotetun informaation sisällön analysointia.

Esitetyn tutkimuksen pohjalta on selkeästi osoitettavissa muutamia jatkokehityssuuntia. Ensinnäkin, digitaaliseen signaalinkäsittelyyn perustuva järjestelmä on sovellettavissa myöskin muuntyyppisten kaukokirjoitus- ja datalähetteidien vastaanottoon. Erityisen mielenkiintoisia kehityssuuntia tarjoavat vaiheensiirtoavainnukseen (PSK) tai useamman äänitaajuuden vaihtotaajuusavainnukseen (MFSK) perustuvien menetelmien tutkiminen. Tämän tyyppiset lähetyslajit ovat viime vuosina yleistyneet ja tarjoavat uusia mahdollisuuksia entistä varmempaan ja yleensä myös nopeampaan datan siirtoon heik-

kojakin signaaleja käyttäen. Toinen mielenkiintoinen jatkokehityssuunta on kaukokirjoitusläheteiden tunnistaminen ja analysointi niiden sisällön perusteella. Myös vastaanottojärjestelmän kaukokäyttömahdollisuuksien tutkiminen tarjoaa jatkokehityskohteita.

Viiteluettelo

- [Arias, 2001] Jesús Arias, RTTY: an FSK decoder program for Linux, 2001.
Available as <http://www.ele.uva.es/~jesus/rtty/rtty-2.0/doc/rtty.pdf> 10.8.2001.
- [ARRL, 2000] *Technical Descriptions: CLOVER, CLOVER-2000, G-TOR, PACTOR, PACTOR II, PSK 31*. American Radio Relay League, Inc., Newington, CT, 2000.
- [Carlson *et al.*, 2002] A. Bruce Carlson, Paul B. Crilly and Janet C. Rutledge, *Communication Systems, Fourth Edition*. McGraw-Hill, New York, NY, 2002.
- [Cormen *et al.*, 1997] Thomas H. Cormen, Charles E. Leiserson and Ronald L. Rivest, *Introduction to Algorithms*. The MIT Press, Cambridge, MA, 1997.
- [DWD, 2002] Deutscher Wetterdienst, RTT broadcast for shipping.
Available as http://www.dwd.de/services/gfsf/e_telexpln.htm 14.5.2002.
- [Engdahl, 1993] Tomi Engdahl, RS-232C-sarjaliitäntä IBM PC/AT:ssa ja yhteensopivissa. Available as <http://www.hut.fi/~then/mytexts/rs-232c.html> 3.3.2001.
- [Exar, 1992] Exar, *XR-2211 FSK Demodulator / Tone Decoder, Data Sheet, rev. 3.01*, Exar Corporation, 1992.
- [Fisher, 1999] Tony Fisher, Interactive digital filter design,
Available as <http://www-users.cs.york.ac.uk/~fisher/mkfilter> 15.4.2002.
- [Flyktman *et al.*, 1994] Reima Flyktman, Tapani Lahtinen, Kimmo Rousku, Timo Rousku ja Jyrki Kivimäki, *PC-mikron laitekirja 1*. Suomen ATK-kustannus Oy, 1994.
- [Ford, 1999] Steve Ford, *ARRL's HF Digital Handbook*. The American Radio Relay League, Inc., Newington, CT, 1999.
- [Forrer, 1996] Johan Forrer, A DSP-Based Audio Signal Processor, *QEX* **176** (September 1996), 8-13.

- [Goldt *et al.*, 1995] Sven Goldt, Sven van der Meer, Scott Burkett and Matt Welsh, *The Linux Programmer's Guide, version 0.4*. Linux Documentation Project, 1995. Available as <http://www.tldp.org/LDP/lpg/index.html> 23.05.2002.
- [GPL, 1991] GNU General Public License, Version 2, Free Software Foundation, Inc., Boston, MA, June 1991. Available as <http://www.gnu.org/licenses/gpl.txt> 24.05.2002.
- [Helfert and Strate, 1991] Hans-Peter Helfert and Ulrich Strate, PACTOR – Radioteletype with Memory ARQ and Data Compression, *QEX* **148** (October 1991), 3-6.
- [Jackson, 1986] Leland B. Jackson, *Digital Filters and Signal Processing*. Kluwer Academic Publishers, Norwell, MA, 1986.
- [Lane and Hillman, 1993] John Lane and Garth Hillman, *Implementing IIR/FIR Filters with Motorola's DSP56000/SPS/DSP56001 Digital Signal Processors, Rev. 2*. Motorola Inc., Digital Signal Processing Division, 1993.
- [Lynn and Fuerst, 1998] Paul A. Lynn and Wolfgang Fuerst, *Digital Signal Processing with Computer Applications, Second Edition*. John Wiley and Sons, Chichester, 1998.
- [Maxim, 1996] Maxim, *15kV ESD Protected, +5V RS-232 Transceivers, Data Sheet, Rev. 3*, Maxim Integrated Products, 1996.
- [Morris, 1985] John Morris, *Amateur Radio Software*. Radio Society of Great Britain, Hertfordshire, 1985.
- [Peterson and Weldon, 1981] W. Wesley Peterson and E.J. Weldon, Jr., *Error Correcting Codes, Second Edition*. The MIT Press, Cambridge, MA, 1981.
- [Press *et al.*, 1997] William H. Press, Saul A. Teukolsky, William T. Vetterling and Brian P. Flannery, *Numerical Recipes in C - The Art of Scientific Computing, Second Edition*. Cambridge University Press, New York, NY, 1997. Partially available at <http://www.nr.com> 10.05.2002.
- [Renton, 1974] R.N. Renton, *The International Telex Service, First Edition*. Pitman Publishing, London, 1974.

- [Rink and Helfert, 1995] Dr. Tom Rink and Hans-Peter Helfert, PACTOR-II – The New Dimension in Data Transmission Technology. Originally published in the *ADRS Digital Journal* (1995). Available as <http://www.snowy.net.au/srs/ptc2tch.htm> 10.03.2002.
- [SCS, 2002] Special Communication Systems GmbH, *Pactor III, the ultimate speed experience*. Available as <http://www.scs-ptc.com/pactor3.html> 24.03.2002.
- [Seewetter, 2002] Seewetter-Info, *SYNOP Code – FM 12: Surface Synoptic Observations*. Available as <http://www.seewetter-info.de/fm12.html> 14.05.2002.
- [Smith, 1999] Steven W. Smith, *The Scientist and Engineer's Guide to Digital Signal Processing, Second Edition*. California Technical Publishing, San Diego, CA, 1999. Available as <http://www.DSPguide.com> 10.08.2001.
- [Stallings, 1994] William Stallings, *Data and Computer Communications*. Prentice Hall International, London, 1994.
- [Sweet, 1999] Michael R. Sweet, *Serial Programming Guide for POSIX Operating Systems, Fifth Edition*. Available as <http://www.easysw.com/~mike/serial/> 03.03.2001.
- [Taylor, 2001] Joe Taylor, WSJT: New Software for VHF Meteor-Scatter Communication, *QST 85* (December 2001), 36-41.
- [Tranter, 2001] Jeff Tranter, *Open Sound System™ Programmers Guide. Version 1.11*. 4Front Technologies, Culver, CA, 2001. Available as <http://www.opensound.com> 20.01.2002.
- [Wade, 1987] Ian Wade, Amtor primer, part 1. *Radio Communications* **63** (1987), 203.
- [Williams, 1991] Ross N. Williams, *Adaptive Data Compression*. Kluwer Academic Publishers, Norwell, MA, 1991.
- [Woodson, 1975] Charles Woodson, Definition of Coherent CW. *Coherent CW Newsletter* **1** (1975), 1.
- [Yaesu, 1997] Yaesu, *FT-920 Transceiver, Operating Manual*, Yaesu-Musen Co., Ltd., Tokyo, 1997.

Baudot-koodi

	Kirjaintaso	Numerotaso
00000	[BLANK]	[BLANK]
00001	E	3
00010	[LF]	[LF]
00011	A	-
00100	[SPACE]	[SPACE]
00101	S	'
00110	I	8
00111	U	7
01000	[CR]	[CR]
01001	D	[WRU]
01010	R	4
01011	J	[BELL]
01100	N	,
01101	F	! tai Å
01110	C	:
01111	K	(
10000	T	5
10001	Z	+ tai "
10010	L)
10011	W	2
10100	H	\$ tai Ö
10101	Y	6
10110	P	0
10111	Q	1
11000	O	9
11001	B	?
11010	G	& tai Ä
11011	[FIGS]	[FIGS]
11100	M	.
11101	X	/
11110	V	= tai ;
11111	[LTRS]	[LTRS]

XR-2211 piirillä toteutetun demodulaattorin laskennalliset sekä valitut komponenttiarvot

	45,45 baudia, $\Delta f = 170 \text{ Hz}$ $f_1 = 1275 \text{ Hz}$ $f_2 = 1445 \text{ Hz}$	50 baudia, $\Delta f = 450 \text{ Hz}$ $f_1 = 1135 \text{ Hz}$ $f_2 = 1585 \text{ Hz}$	45,45 baudia, $\Delta f = 170 \text{ Hz}$ $f_1 = 2125 \text{ Hz}$ $f_2 = 2295 \text{ Hz}$	50 baudia, $\Delta f = 450 \text{ Hz}$ $f_1 = 1985 \text{ Hz}$ $f_2 = 2435 \text{ Hz}$	Valitut komponentti- arvot
Keski- taajuus	$F_0 = 1360 \text{ Hz}$		$F_0 = 2210 \text{ Hz}$		Säädettävissä 1280 ... 2560 Hz
C_0	39 nF	39 nF	39 nF	39 nF	39 nF
C_1	650 pF	1,7 nF	650 pF	1,7 nF	1 nF
C_F	3,8 nF	8,7 nF	3,8 nF	8,7 nF	4,7 nF
C_D	> 12 nF	> 12 nF	> 7 nF	> 7 nF	47 nF
R_0	18854 Ω	18854 Ω	11602 Ω	11602 Ω	10k Ω +10k Ω (pot.)
R_1	301664 Ω	113962 Ω	301652 Ω	113957 Ω	220 k Ω
R_B	> 7,5 M Ω	> 3 M Ω	> 7,5 M Ω	> 3 M Ω	4,7 M Ω
R_F	> 1,5 M Ω	> 600 k Ω	> 1,5 M Ω	> 600 k Ω	1 M Ω
R_L	4,7 k Ω	4,7 k Ω	4,7 k Ω	4,7 k Ω	4,7 k Ω
R_D	470 k Ω	470 k Ω	470 k Ω	470 k Ω	470 k Ω

