

Juha Keränen

Tiedonsiirto ennustamalla

Tietotekniikan pro gradu -tutkielma

12. tammikuuta 2017

Jyväskylän yliopisto

Tietotekniikan laitos

Tekijä: Juha Keränen

Yhteystiedot: juha.t.keranen@student.jyu.fi

Ohjaaja: Viinikainen, Ari

Työn nimi: Tiedonsiirto ennustamalla

Title in English: Data transfer by predicting

Työ: Pro gradu -tutkielma

Suuntautumisvaihtoehto: Tietoliikenne

Sivumäärä: 62+0

Tiivistelmä: Tiedon tuottaminen ja siirto on tulevaisuudessa merkittävässä osassa kaikkea verkkoliikennettä. Ennusteiden mukaan sekä laitteiden että kerätyn tiedon määrä moninkertaistuu tulevaisuudessa. Tämä aiheuttaa haasteita laitteille ja tiedon siirtoverkoille. Tässä tutkielmassa selvitetään, voidaanko tiedon ennustamisella säästää resursseja niin siirtokanavien kuin laitteiden osalta. Ennustamisessa hyödynnetään konetunnistamisen menetelmiä ja mallia simuloidaan oikealla tiedolla.

Avainsanat: Tiedonkeruu, tiedonkäsittely, tiedonhallinta, ennustaminen, signaalinsiirto

Abstract: Production and transfer of information will constitute a substantial part of all network traffic in the future. According to estimates, the amount of equipment as well as the amount of information gathered will multiply. This creates challenges for both the equipment and the data transfer networks. This study explores the possibilities of data predicting in saving resources concerning the transfer networks as well as the equipment. In creating the predicting model the machine learning methods are utilized and the model is simulated using real world data.

Keywords: Data collection, data processing, data management, predicting, signal transfer

Esipuhe

"Kandista maisteriksi 11 kuukaudessa" - projekti ei olisi voinut onnistua tämän paremmin.

Kiitoksia perheelleni, joka on joutunut venymään kiireideni mukaan. Erityiskiitos ohjaajaleni, ilman häntä tämä ei olisi onnistunut tässä aikataulussa.

Tutkielma oli prosessina haasteellinen. Aihetta on tutkittu, mutta tutkielman näkökulmasta ei löytynyt soveltuvaa materiaalia. Tutkielma aihe on kuitenkin ajankohtainen, joten uteliaisuus pakotti jatkamaan.

"Miten ajattelen, jos en ajattele toisin

Mihin häkkiin lensinkään

Miten ajattelen, jos en ajattele toisin

Ajattelenko ensinkään"

- Juice Leskinen

Jyväskylässä 3.1.2017

Juha Keränen

Kuviot

Kuvio 1. Signaalin käsittely.....	5
Kuvio 2. Kommunikointijärjestelmä (mukaellen Shannon 2001, s. 4).....	6
Kuvio 3. Modbus arkkitehtuuri (mukaellen Modbus Organization 2006, s. 4).....	10
Kuvio 4. Modbus/TCP kommunikointi.....	10
Kuvio 5. Profinet arkkitehtuurit (mukaellen PROFIBUS and PROFINET International. 2016, s. 1)	12
Kuvio 6. Profinet tosiaikainen tiedonsiirto (Siemens 2008, s. 17)	13
Kuvio 7. MPEG periaate (mukaellen Le Gall 1991, s. 52)	14
Kuvio 8. RLE koodin toiminta.....	14
Kuvio 9. LZW koodaus (mukaellen Welch 1984, s. 16).....	15
Kuvio 10. LZW dekkoodaus (mukaellen Welch 1984, s. 16).....	16
Kuvio 11. Paloittan määritellyn funktion kuvaaja	17
Kuvio 12. Liukuva ikkuna	17
Kuvio 13. Tutkielmassa käytetty luokittelu päätöksentekopuussa	19
Kuvio 14. Kirjallisuudessa esiintyvät aikasarjaesitykset (Lin ym. 2012, s. 8)	20
Kuvio 15. Liukuvan ikkunan menetelmällä tiivistys (mukaellen Lin ym. 2003, s. 7).....	21
Kuvio 16. Ennustava kommunikointijärjestelmä	25
Kuvio 17. Signaalin tunistaminen ja lähetyspisteet	28
Kuvio 18. Kulmakertoimen laskeminen	29
Kuvio 19. Kanttiaallon mallinnus	31
Kuvio 20. Sinisignaalin mallinnus.....	32
Kuvio 21. Linearisesti muuttuvan signaalin mallinnus	33
Kuvio 22. Testijärjestelmä	35
Kuvio 23. Ennustava testi protokolla	36
Kuvio 24. Lähettävän ohjelman toiminta.....	37
Kuvio 25. Sanomarakenne	38
Kuvio 26. Vastauksen ennustaminen	39
Kuvio 27. Kuvio ennustamistaulukosta Excelissä	40
Kuvio 28. Modbus/TCP- kommunikointi (mukaellen Modbus Organization 2006, s. 17)..	42
Kuvio 29. Modbus sanomaliikennettä	43
Kuvio 30. Kanttiaallon siirtäminen ennustamalla.....	43
Kuvio 31. Siniaallon siirtäminen ennustamalla	44
Kuvio 32. Lineaarisen signaalin siirtäminen ennustamalla.....	45

Taulukot

Taulukko 1. <i>Paloittain määritellyt aikasarja algoritmit (mukaellen Keogh ym. 2001, s. 292).</i>	20
Taulukko 2. <i>Kanttiaallon siirron ennustus.</i>	43
Taulukko 3. <i>Siniaallon siirron ennustus.</i>	44

Taulukko 4. <i>Lineaarisesti muuttavan signaalin siirron ennustus</i>	45
Taulukko 5. <i>Auringonsäteilysignaalin siirron ennustus</i>	46

Sisältö

1	JOHDANTO	1
2	MITTAAMINEN	3
	2.1 Historia	3
	2.2 Mittaaminen	4
	2.3 Informaation siirto.....	5
3	TIEDONSIIRTOTEKNIikka	8
	3.1 Real-Time Ethernet.....	9
	3.1.1 Modbus/TCP	9
	3.1.2 ProfiNet	11
	3.2 Muita mielenkiintoisia tekniikoita tiedon siirtoon ja käsittelyyn	12
	3.2.1 MPEG	12
	3.2.2 RLE ja GIF.....	13
	3.2.3 LZW	15
	3.2.4 Lineaarinen paloittain määritelty funktio	16
	3.2.5 Liukuva ikkuna	17
4	SIGNAALIN TUNNISTUS	18
	4.1 Koneoppiminen signaalin tunnistuksessa	18
	4.2 Aikasarja.....	19
	4.3 Ennustaminen.....	21
	4.3.1 Liukuva keskiarvo	22
	4.3.2 Lineaarinen regressioanalyysi.....	23
5	ENNUSTAVA PROTOKOLLA	24
	5.1 Teoria	24
	5.2 Ennustaminen.....	26
	5.3 Oletuksia.....	29
	5.4 Mallintestaaminen.....	30
	5.4.1 Kanttiaalto	30
	5.4.2 Sinitesti	31
	5.4.3 Linearisesti muuttuva signaali	31
6	TESTIJÄRJESTELYT	34
	6.1 Testijärjestelmä.....	34
	6.2 Testimateriaali	34
	6.3 Ennustava testiprotokolla	35
	6.3.1 Lähetys	36
	6.3.2 Vastaanotto	38
7	TESTI	41
	7.1 Verrokkina Modbus/TCP.....	41
	7.2 Ennustava protokolla	41

7.2.1	Kanttiaalto	42
7.2.2	Sinitesti	44
7.2.3	Lineaarisesti muuttuva signaali	44
7.2.4	Säämittaustieto	45
7.3	Testien ongelmat	47
8	ANALYSOINTI	48
8.1	Kanttiaalto	48
8.2	Sinitesti	48
8.3	Lineaarisesti muuttuva signaali	48
8.4	Säädata	49
9	YHTEENVETO	50
	LÄHTEET	51

1 Johdanto

Ihminen on aina ollut kiinnostunut siitä miten pystyy ilmaisemaan asiat yksiselitteisesti. Eräs tälläisen mahdollistaja on ollut mittayksiköt. Alussa mittaaminen on perustunut ihmisen ominaisuuksiin sekä luonnonilmiöihin. Mittayksiköt ovat siksi liittyneet usein kehon mittoihin tai muuten tuttuihin asioihin. Nykypäivänäkin näiden jäänteitä on useissa maissa, joissa on edelleen käytössä mittayksikköinä jalka, maili ja tuuma. Myös ajan mittaaminen on ollut tuttuihin asioihin sidottua ja vielä nykyäänkin toimitaan esimerkiksi vuoroveden mukaan.

Nykyisin mittaaminen on muuttunut tarkemmaksi ja se liittyy melkein kaikkeen. Mittaaminen on tärkeyttä kuvaa muun muassa se, että VTT (2016) on arvioinut mittaamisen liittyvän yli puoleen Suomen bruttokansantuotteesta. Mittaamisen ja tiedon keräämisen tärkeydestä johtuen tietoa kerätään paljon. Gartnerin (Gartner, Inc. 2014) ennustuksien mukaan kaiken kerätyn tiedon määrä kasvaa viisinkertaiseksi vuosien 2015-2020 välillä, ja tietoa tuottavien laitteiden määrä kasvaa samalla tahdilla. Voikin olettaa datamäärien kasvun aiheuttavan haasteita varsinkin elinkaarensa loppupäässä oleville järjestelmille.

Sen lisäksi, että kerätyn tiedon määrä aiheuttaa ongelmia, myös kerätyn tiedon laatu voi aiheuttaa ongelmia. Perinteisesti tieto on kerätty jotakin käyttöä varten, mutta tulevaisuuden tarpeet voivat olla täysin erilaiset. Tulevaisuudessa tiedon tulee olla monikäyttöistä. Perinteinen tiedon keruutapa ei välttämättä tuota tietoa uusia tarpeita varten, eikä jo kerättyä tietoa pystytä hyödyntämään tehokkaasti. Kun koetetaan löytää uusia käyttötarkoituksia olemassa olevalle tiedolle. Heikosti kerätty tieto voi hukata mahdollisuuksia tiedon hyödyntämisessä.

Tässä Pro Gradu- tutkielmassa keskitytään pelkästään numeerisen tiedon keräämisen ongelmaan. Tutkielmassa perehdytään numeerisen mittaustiedon siirron problematiikkaan ja siihen, miten tietoa saataisiin siirrettyä mahdollisimman pienillä resurseilla niin tiedonsiirtoväylän, mittalaitteen kuin tallenustilan suhteen. Tutkielman tarkoitus on selvittää voidaanko tiedon tunnistamisen avulla mahdollistaa nopeampaa tiedonkeruuta. Selvityksessä käytetään konetunnistamisen keinoja signaalien tunnistamiseen niin signaalia tuottavassa kuin sitä käsittelevässä laitteessa.

Tutkielma koostuu rakenteellisesti neljästä osasta: Ensimmäisessä tutustutaan mittaamiseen

ilmiönä ja ilmiön kehittymistä. Toisessa osassa tarkastellaan erilaisia mittaamiseen, tiedon siirtämiseen ja tiedon käsittelemiseen tarkoitettuja menetelmiä. Kolmannessa osassa suoritetaan testaamista siirtämällä testiympäristössä erilaisia signaaleja, ja viimeisessä osuudessa analysoidaan tuloksia sekä pohditaan, miten tiedonkeruun tulisi kehittyä tulevaisuuden haasteisiin.

Tutkielma on toteutettu ilman sidoksia yrityksiin tai ulkopuolisiin toimijoihin. Tutkielman toteutukset ovat kirjoittajan omaa ideointia ja suunnittelua.

2 Mittaaminen

VTT:n määrittelmän mukaan mittaminen on ominaisuus, joka voidaan laadultaan tunnistaa ja määritellä vertaamalla referenssinä toimivaan yksikköön (VTT 2016). Suomessa mittamisen referenssinä toimivat SI-mittayksiköt. Vaikka SI-yksiköt levinneet ovat varsin laajalle, löytyy myös muita, kuten Yhdysvallat, Burma ja Liberia joissa käytetään virallisina yksiköinä muita kuin SI-yksiköitä (CIA 2016). Mittajärjestelmien eroista huolimatta kaikki mittayksiköt on kuitenkin sidottu johonkin muuttumattomaan ja niille löytyy muunnostaulukot joiden avulla kaikkialla tiedetään, yksiköstä riippumatta, mikä määrä on kyseessä. Yhteisesti tiedetyt mitat ovat tärkeitä ja ilman niitä on vaikeaa, jopa hengenvaarallista, tehdä mitään laajemmassa yhteydessä. BBC on artikkelissaan käynyt läpi kansainvälisiä projekteja, joissa projektit ovat epäonnistuneet mittayksiköistä johtuen, tai aiheuttaneet jopa ihmishenkien menetyksiä (Cheung ja Mulvey 2014).

2.1 Historia

Mittaaminen on ollut tärkeä osa elämistä ja kreikkalainen filosofi Protagoras (481-42 eaa.) on muun muassa sanonut:

”Kaikkien arjen asioiden tarpeen mitta on ihminen, sellaisina kuin asiat ollessaan ovat, ja kun niitä ei ole, myös silloin”

Vaikka Protagorasin sanonnan asiayhteys on aikojen saatossa kadonnut ja voimme vain arvailla niitä kielellisesti (esimerkiksi Heidegger (Heidegger 1977, s. 19-23)), voimme olettaa mittamisen olleen tärkeää Kreikkalaisten filosofien keskuudessa, koska sitä on käytetty filosofisen ajattelun argumentoinnin apuna.

Kreikan filosofiasta huolimatta nykyaikaisen mittamisen juuret tulevat ranskasta, tarkemmin ranskan vallankumouksen ajalta. Vallankumousta edeltävässä Ranskassa oli käytössä yli 700 erilaista mittausta esittävää suuretta, ja se aiheutti vaikeuksia tietessä ja teollisuudessa. Käytetyt suuret olivat riippuvaisia ihmisten eri ruumiinosista ja niiden ominaisuuksista, ja tästä syystä niiden vaihtelutkin olivat suuret, koska referenssit vaihtuivat käyttäjien mu-

kaan. Tieteen sekä teollisuuden tarpeet olivatkin sellaiset että mittayksiköt piti saada muuttumattomiksi ja tieto siirrettyä tarkemmin kohteiden välillä. Ranska julkaisikin 22.6.1799 kaksi mittayksikköä, metrin ja kilogramman, selventämään ongelmaa. (LNE 2016) Näistä valmistettiin referenssit joiden avulla saatiin yhteneviä malleja eripuolille maapalloa, ja siten yhteiset mittayksiköt levisivät yleiseen käyttöön.

Vaikka nykyisin mittayksiköt on sidottu luonnonvakioihin, on yhden kilogramman platinasta ja iridiumista valmistettu prototyyppi, ”Le Grand K”, edelleen ainoa virallinen referenssi ja sitä säilytetään Pariisissa kansainvälisessä painojen ja mittojen toimistossa BIPM:ssä (Bureau International des Poids et Mesures) (BIPM 2016).

2.2 Mittaaminen

Mittaaminen on muuttunut vuosien saatossa tekniikan kehittyessä. Kuten muussakin tekniikassa on elektorniset apuvälineet tulleet avuksi ja nykyaikainen mittaaminen tapahtuu usein digitaalisella mittalaitteella, ja usein saatua mittaustulosta prosessoidaan ennen kuin se esitetään.

Periaatteeltaan mittaminen digitaalisesti on kolmivaiheinen prosessi joita ovat (Orfanidis 1995, s. 1):

1. Analoginen viesti digitalisoidaan
2. Digitalisoidut näytteet käsitellään signaaliprosessorilla
3. Digitaalinen viesti muunnetaan takaisin analogiseksi

Signaaliprosessointia on muun muassa suodattaminen. Sitä tehdään melkein kaikelle signaalille häiriöiden poistamiseen.

Kuviona (Kuvio 1) digitaalinen signaalin käsittely on melko selkeä. Mittaus luetaan, muunnetaan digitaalseksi, prosessoidaan ja esitetään.

Jos mittaamista ajatellaan verkon yli tapahtuvaksi, niin signaalin prosessoinniksi voidaan käsittää siirtäminen verkon yli. Mittaustieto muunnetaan digitaaliseen muotoon ja sitä käsitellään kunnes se sitten on valmis ja perillä, ja voidaan muuntaa taas käytettävään muotoon.



Kuvio 1. Signaalin käsittely

2.3 Informaation siirto

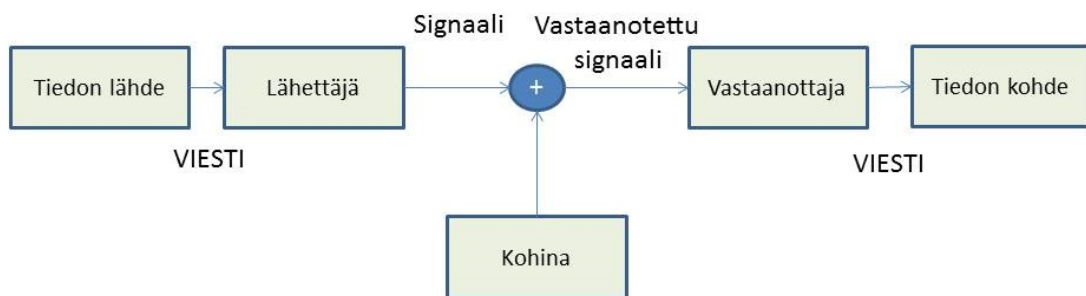
Että tieto olisi käytettävissä muuallakin kuin tietoa tuottavassa kohteessa, se tulee voida siirtää paikasta toiseen. Tiedon siirtämiseen on käytetty mitä moninaisempia keinoja. Jos mietitään historiallisia keinoja siirtää tietoa, tulee mieleen esimerkiksi intiaanit savumerkkeineen ja laivat merkkilippuineen. Näistä on jossakin vaiheessa edetty lennättimen ja puhelimen kautta nykyiseen digiyhteiskuntaan, jossa tietoa on mahdollista siirtää paljon ja useilla eri tavoilla. Tietoa on siirretty enemmän ja enemmän, ja se onkin muuttunut jokapäiväiseksi osaksi elämäämme.

Moni tutkija mietti tiedonsiirron periaatteita 1900 -luvulla. Tunnetuin näistä on Claude Shannon (1916–2001), joka 1940-luvulla kehitti teoriaa tiedon luotettavasta mittaamisesta ja siirtämisestä. Vuonna 1948 hän julkaisi artikkelin *A Mathematical Theory of Communications* (Shannon 2001), jossa hän käsitteli aihetta ja esitteli kuuluisaksi tulleet kaksi teoreemaansa. Näistä tunnetuimmista teorioista ensimmäinen (Shannon 2001, s. 35) käsittelee tiedon pakkaamisen ongelmaa eli sitä, miten pienessä tilassa tieto voidaan esittää hukkaamatta informaatiota. Toisessa teoriassa Shannon esittää, miten paljon tietoa voidaan siirtää tiedonsiirtokanavassa (Shannon 2001, s. 3–32).

Nämä teoreemat ovat olleet keskeisin ajatuksin niin tiedonsiirron kuin pakkaustekniikoidenkin perusteorioita, ja voi väittää että niiden vaikutukset ovat olleet merkittäviä nykyisen tietoyhteiskunnan kannalta. Yksin Shannon ei kuitenkaan näihin ajatuksiin päätenyt vaan niihin tuloksiin hän tuli jatkokehittäessään edelleen Hartleyn (Hartley 1928) ja Nyquistin (Nyquist 1928) teorioita. Siksi näitä teorioita kutsutaankin usein myös Shannon–Hartley- ja Nyquist–Shannon–teoreemoiksi.

Samoja aiheita tutki ja samoihin johtopäätöksiin Shannonin kanssa päätyi moni muukin tutkija, kuten Edmund Whittaker ja Vladimir Kotelnikov. Tästä syystä muunumuassa näytteenottoteoreema tunnetaan usein muunkin henkilön nimellä. (Jerri 1977, s. 1564)

Shannonin tiedonsiirtojärjestelmä on yksinkertaisuudessaan kuvion mukainen (Kuvio 2). Shannonin pyrkimyksenä oli keksiä keinoja tiedonsiirron kysymyksiin vääristymien estämiseksi, ja hänen tutkimuksissaan toistui häviämättömyys ja toistettavuus. Miten tieto voidaan siirtää muuttumattomana ja mahdollisimman nopeasti paikasta toiseen siten, että se saadaan palautettua riittävällä tarkkuudella.



Kuvio 2. Kommunikointijärjestelmä (mukaellen Shannon 2001, s. 4)

Shannonin kommunikointijärjestelmän mallissa tuotetaan tietoa, joka voi olla joko merkkejä tai funktioita tai niiden yhdistelmiä, eli tieto voi olla muodoltaan hyvin erilaista (Shannon 2001, s. 4). Käytännössä Shannonin tieto on kaikkea tietoa, jota voi muuntaa sähköisesti siirrettäväksi. Yksinkertaistettuna voitaneen ajatella Shannonin siirtotie häiriönä, joka rajoittaa tiedonsiirtoa, ja jonka seasta vastaanottaja koettaa saada selvää lähetetystä tiedosta. Tästä vastaanotetusta tiedosta viesti rakennetaan uudelleen ja siirretään ymmärrettävässä muodossa eteenpäin kohteeseen.

Tämä tutkielman tarkastelun näkökulma on pitkälti yhtenevä Shannonin kommunikointijärjestelmän kanssa. Se pohtii miten siirretään tietoa luotettavasti ja suurella nopeudella, samalla mahdollisimman vähän tiedon siirtokanavaa ja laitteita kuormittaen. Shannonin ajatuksista poiketen tutkielma selvittää onko tietoa mahdollista ennustaa ennalta määritellyn mallin perusteella, jolloin sekä lähettjä että vastaanottaja pyrkivät tunnistamaan siirrettävän tiedon ja laskemaan ennustettavissa olevat arvot. Shannonin järjestelmän häiriöt jäävät kokonaan pois ennustettavasta tiedosta. Näin siirtotien kapasiteettia saadaan kasvatettua. Shannonin ja digitaalisen mittauksen käsittelyn ajatukset ovat yhteneviä, kun vertaa siirtojärjestelmien rakennetta Shannonin (Kuvio 2) ja digitaalisen mittauksen käsittelyn (Kuvio 1) osalta. Mit-

tauksen prosessointi on signaalin siirtoa ja tutkielman kannalta hyvä testi ympäristö.

3 Tiedonsiirtotekniikka

Tiedonsiirtoa tapahtuu kaikkialla ja kaiken aikaa. Televisio, tietokone, puhelin, lehdet ja kirjat tekevät sitä, eikä aina tule ajatelleeksi, että pohjimmiltaan niiden kaikkien perimmäinen toiminta on sama, tiedonsiirto kahden tai useamman kohteen välillä. Kommunikointitapoja on monenlaisia ja kommunikointi voi tapahtua usealla eri tavalla. Mutta pelkästään tiedonsiirtotekniikka ei tee onnelliseksi, vaan myös viestin tulee olla ymmärrettävä. Se riippuu kulttuurista, sosioekonomisesta taustasta, koulutuksesta yms.. Vaikka muuten kaikki täsmää, voi yhteinen kieli silti puuttua.

Kommunikointi on kuitenkin niin tärkeää, että ihmiset ovat keksineet erilaisia menetelmiä helpottamaan kommunikointia. Tästä hyvänä esimerkkinä on Esperanto, keinotekoinen kieli, joka on kehitetty ihmisten kommunikoinnin helpottamiseksi.

Jos tiedon onnistunut siirtäminen on ihmisten välisessä kommunikoinnissa on tärkeää, on se tärkeää myös ihmisten käyttämien laitteiden välillä. Koneiden väliseen kommunikointiin suunniteltujen kielten sääntöjä kutsutaan protokolliksi. Kurose ja Ross (2013, s. 35) määrittelevät protokollan seuraavasti:

"A protocol defines the format and the order of messages exchanged between two or more communicating entities, as well as the actions taken on the transmission and/or receipt of a message or other events."

Protokolla on siis kahdenvälinen sopimus siitä miten vaihdetaan viestejä. Protokolla pitää olla tarkkaan määritelty, jotta tieto osataan tulkita vastaanottavassa päässä oikein. Koneet eivät vielä kykene tulkitsemaan kuten ihmiset, ja tästä syystä koneiden tulee tarkasti noudattaa yhteisesti sovittuja käytäntöjä kommunikoinnin onnistumisen takaamiseksi. Tulevaisuudessa IBM:n Watsonin kaltaiset järjestelmät tulevat muuttamaan tätäkin asiaa (**ferrucci2010building**), mutta tällä hetkellä tiedonvaihdon on edettävä yhteisten sääntäjien mukaan ilman tulkintaa.

3.1 Real-Time Ethernet

Real-Time Ethernet on nimensä mukaisesti tosiaikaista kommunikointia. Tosiainen on tosin nimenä hiukan harhaanjohtava koska tiedonsiirrossa tapahtuu aina viiveitä. Se miten tosiaikaisuus määritellään on jo vaikeampi asia. Kopetzin (Kopetz 1991, s. 87) mukaan järjestelmä määrittelee tosiaikaisuuden vaatimukset aikarajalla ja mikäli järjestelmä ei pysty toimimaan aikarajassa, ei se ole tosiaikainen. Felser (2005, s. 1212) esittääkin kolmea aikakriteeriä tosiaikaisuuden luokitteluun eri tehtävissä. Näitä ovat 100ms, 10ms ja 1 ms. Neumann (2007, s. 1333) antaa näille tarkemmat vaatimukset, joissa nopeimmillaan toimitaan $250\mu s$ aikasyklillä.

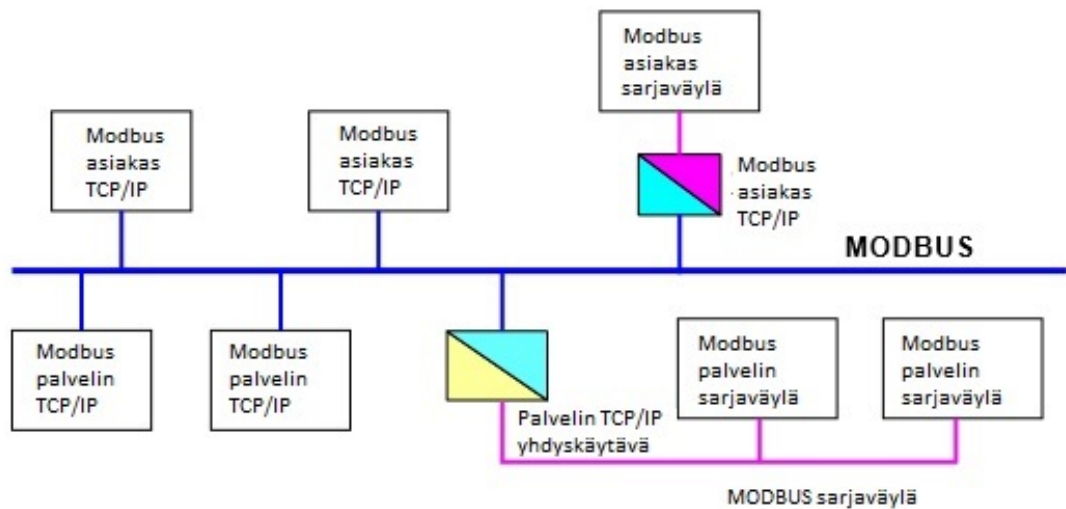
Kopetz lisäksi erottaa tosiaikaisen tiedonsiirron rakentamisen kahteen eri tekniikkaan, tapahtumapohjaiseen (engl. event triggered) ja aikaperusteiseen (engl. time triggered) tiedonsiirtoon. Tapahtumapohjaisessa tiedonsiirrossa siirto tapahtuu tapahtuman perusteella, ja aikaperusteisessa tietoa siirretään tietyllä aikasyklillä (Kopetz 1991, s. 90-95). Näistä aikapohjainen on useimmin käytössä yksinkertaisemman toteutuksensa vuoksi (Decotignie 2005, s. 1103).

Tässä tutkielmassa keskitytään mittaustiedon siirtoon ja siksi tarkastellaan kahta teollisuuden parissa *"de-facto"* -asemaan noussutta protokollaa. Modbus ja Profibus ovat eri versioineen hyvin suosittuja, ja siksi niitä ei voi ohittaa mittaus tiedonsiirrosta puhuttaessa.

3.1.1 Modbus/TCP

Modbus (Modbus Organization 2017) on alkujaan Modiconin vuonna 1979 julkaisema sarjaliikenneprotokolla. Protokolla kehitettiin Modicon omiin tarpeisiinsa siirtämään sarjaliikenteellä tietoa ohjelmoitavien logiikoiden välillä. Protokollasta tuli kuitenkin niin suosittu että muutkin valmistajat rupesivat sitä käyttämään, ja Modbus on tullut *"de-facto"* -standardiksi teollisuudessa. Voikin sanoa ettei ole laitevalmistajaa, jonka laitteita ei saisi liitettyä Modbusin avulla toisiinsa. Nykyisin Modbusista on olemassa sarjaliikenneversion lisäksi myös TCP/IP:n päälle rakennettu versio, joka tukee nykyaikaisempia verkkoratkaisuja. TCP/IP:n päälle rakennettu protokolla tukee sarjaliikenteellä rakennettuja väyliä, joten protokolla on yhteensopiva linkin kautta myös vanhempiin sarjaliikennepohjaisiin Modbus-järjestelmien

kanssa (Kuvio 3).



Kuvio 3. Modbus arkkitehtuuri (mukaellen Modbus Organization 2006, s. 4)

Modbusin kyselyrakenne on asiakas/palvelin-arkkitehtuurin mukainen, eli asiakas kysyy ja palvelin vastaa (Kuvio 4).



Kuvio 4. Modbus/TCP kommunikointi

Kommunikoinnin eteneminen (Modbus Organization 2006, s. 2):

1. Asiakas kysyy palvelimelta tietoa
2. Palvelin ottaa kyselyn vastaan ja käsittelee kyselyn
3. Palvelin vastaa asiakkaalle
4. Asiakas ottaa vastauksen vastaan.

Kommunikointi tapahtuu aina ensin asiakkaasta palvelimeen päin. Asiakas lähettää viestin jossa kertoo mitä tietoja haluaa, ja palvelin lähettää tiedon paluuviestinä. Kyselyn rakenne perustuu palvelimen rekisteripaikkoihin eli kysyjän tulee tietää mitä tietoa hakee ja mistä

rekisteripaikasta se löytyy. Kysymys sisältää IP-osoitteen, ala-asematunnuksen, kyselyfunktion, kysyttävän rekisterialueen aloitusosoitteen sekä kysyttävien rekisterien määrän. Vastauksena tulee rekisterien sisältöjä, joista kysyjä rakentaa kysyttävät arvot.

Asiakas/palvelin ajattelumalli on Modbusin kohdalla hieman sekava verrattuna joihinkin muihin järjestelmiin. Sekavuus johtuu siitä kun Modbus/TCP:ssä tietoa kysyvä asiakas hallinnoi verkkoa ja on oikeastaan palvelin, ja tietoa tuottava palvelin vaikuttaisi olevan enemmän asiakas kuin palvelin. Usein törmääkin tilanteeseen, jossa eri valmistajat käyttävät termejä ristiin ja pelkän nimen perusteella ei voi olla varma tarkoitetaanko asiakasta vai palvelinta.

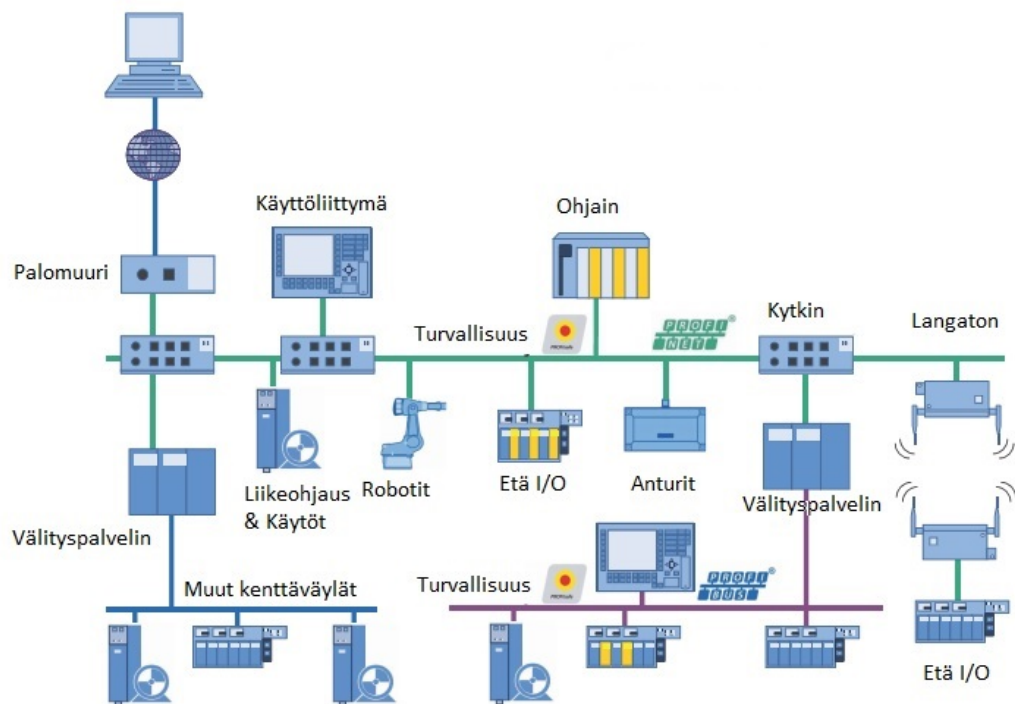
3.1.2 ProfiNet

Toinen yleisesti ja laajasti levinnyt kommunikointi standardit ovat Profibus ja Profinet. Ne ovat avoimen standardin protokollia jotka soveltuvat niin teollisuuden kuin kiinteistötekniikan alalle, ja ovat muutenkin levinneet hyvin laajalle käyttösektorille. Näistä Profibus on sarjaliikenteeseen perustuva tiedonsiirtoprotokolla ja Profinet teollisuus-ethernetiin pohjautuva protokolla. PROFIBUS & PROFINET International (PI) vastaa näiden kummankin ylläpidosta ja kehitysvastuussa. (PROFIBUS and PROFINET International. 2016, s. 1)

Profinet perustuu Ethernetiin, tukee se TCP/IP:tä ja voidaan kätevästi yhdistää muun verkon kanssa (Kuvio 5).

Profinetin soveltuvuus laajasti erilaisiin tarkoituksiin perustuu mahdollisuuteen toimia sekä synchronisena tosiaikaisena tiedonsiirtoverkkona, että perinteisenä asynchronisena tiedonsiirtoverkkona. Tosiakaisuus on toteutettu tekniikalla, jossa liikennöintiverkko synkronoidaan yhteiseen aikaan ja tietoa tuottava laite lähettää määriteltyä tietoa kellon tahdistamana. Vastanottaja tarkkailee saapunutta liikennettä ja odottaa tietoa tulevan määritellyllä aikasyklillä ilman erillistä kommunikointia. (PROFIBUS and PROFINET International. 2016, s. 6–10) Kuten kommunikoinista voi päätellä, vaatii tosiaikaisuus laitteiden tuen tiedonsiirtoon.

Samassa tiedonsiirtoverkossa voi toimia laitteita jotka tulee tosiaikaisuutta, ja laitteita jotka toimii yleisempien protokollien kanssa. Tämä on mahdollista vuorottelun ansiosta. Vuorottelulla tarkoitetaan kun eri aikaikkunoissa lähetetään eri tekniikalla toteutettua tietoa (Kuvio



Kuvio 5. Profinet arkkitehtuurit (mukaellen PROFIBUS and PROFINET International. 2016, s. 1)

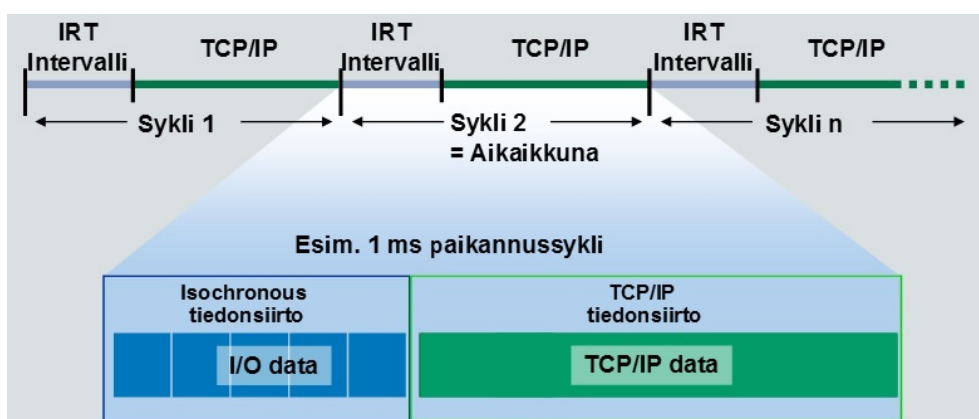
6).

Profinet käyttämä perinteinen TCP/IP on käytössä muun muassa määrittelyssä ja diagnostiikassa, missä ei vaadita tosiaikaisuutta ja tosiaikainen sitten IO-tiedon siirrossa sekä kommunikoinnin monitoroinnissa.

3.2 Muita mielenkiintoisia tekniikoita tiedon siirtoon ja käsittelyyn

3.2.1 MPEG

MPEG (engl. Moving Picture Experts Group) (MPEG 2017) on kuvan ja äänen pakkaamisen kansainvälisten standardien kehittämiseen erikoistunut ryhmä. Heidän kehittämiä standardeja on jo yli toistakymmentä. Tuskin kukaan on välttynyt kuulemasta MPEG koodekeista tai MP3:sta (MPEG-1 Audio Layer 3). Le Gall (1991) esittelee sen yhtenä tärkeimpänä merkki-paaluna digitaalisen videotekniikan osalta, eikä suotta. Hän esittää MPEG:n tehdyn tarkoin harkitusti täyttämään multimedian tarvetta saada yhtenäinen standardi äänen ja kuvan siir-



Kuvio 6. Profinet tosiaikainen tiedonsiirto (Siemens 2008, s. 17)

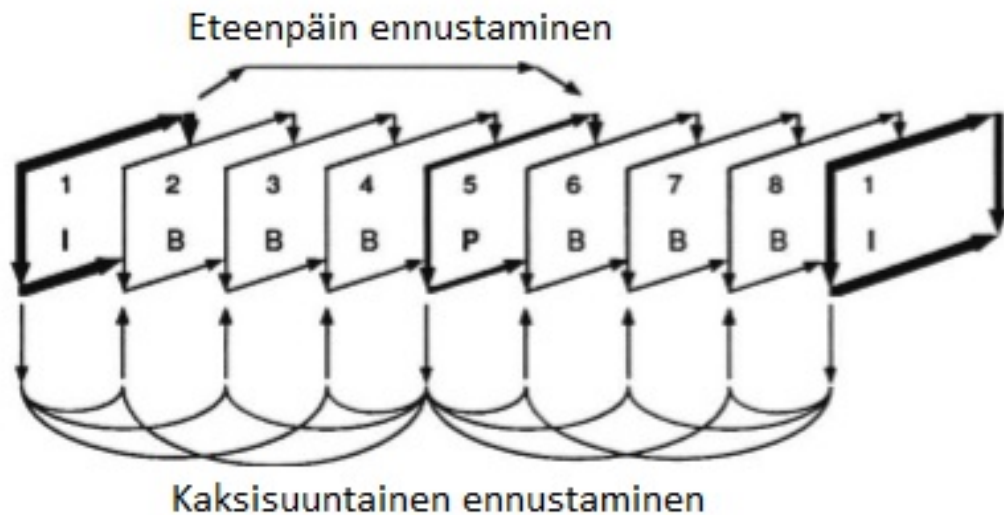
toon. Määrittelytyöhön perustettiin vuonna 1988 työryhmä johon osallistui viisioista jäsentä. Nykyisin mittakaava on kasvanut huomattavasti alkuajoista, ja tapaamisiin osallistuu jo 350 asiantuntijaa 200:sta yrityksestä. (MPEG 2017)

Jos tarkastellaan ensimmäistä pakkaustandardia MPEG-1:tä tarkemmin, sen teho perustuu muuttuvan tiedon esittämiseen ja ennustamiseen. MPEG-1 koostuu kolmenlaisista I- (perus, engl. intra), P- (ennustettu, engl. predicted) ja B- (kaksisuuntainen, engl. bidirectional) koodatuista kentistä (Kuvio 7). Näistä I-kenttä on koodattu sisäisesti ja sisältää eniten tietoa. B-kenttä lasketaan edellisistä ja seuraavista kentistä ja sisältää vähemmän tietoa, mutta vaatii eniten laskentakapasiteettia. P-kenttä lasketaan ennustamalla I:n ja B:n kentän perusteella, joten siihen tietoa ei siirretä suoraan ollenkaan. (Järvinen 2003, s. 432-433; Le Gall 1991, s. 51-53)

MPEG- pakkaustekniikan tehokkuus siis perustuu siihen, että osa kuvien tiedoista voidaan mallintaa jo siirretyn tai lasketun tiedon perusteella, eikä sitä siirretä ollenkaan pakkaamisen ja purkamisen välillä. Kumpikin tietää miten kuva muodostetaan, ja siksi sitä ei tarvitse siirtää.

3.2.2 RLE ja GIF

Juoksupituuskoodaus (engl. run length encoding) koodaa toistuvat rakenteet häviättömästi. CompuServe kehitti sen alkuaan säätutkakuvien siirtoa varten. Sen toiminta perustui sii-



Kuvio 7. MPEG periaate (mukaellen Le Gall 1991, s. 52)

hen, että se tiivistä kaksiväristen kuvien sisällön muotoon, jossa ne oli helppo siirtää ja avata muissa järjestelmissä. Koodauksen ideana on kertoa toistuvan rakenteen alussa montako kertaa asia toistuu. Muun muassa FBI-käytti omien etsityimpien listan kuvien kanssa tätä tekniikkaa. (Dunn 1986, s. 16-17)

Nykyisin pakkausta käytetään muutenkin kuin pelkästään kahden pikselin värin kerrontaan. Pakkauksen tekniikka on helppoa esittää esimerkillä.

Esimerkki 1.

Merkkijono on "AAAAAAAAVVVVVVVDDDDFFFDDDDDDDDDDDDDDDDDDDD". Tästä selvitetään toistuvat merkit ja annetaan merkkien määrä koodatussa merkkijonossa toistuvan rakenteen sijasta (Kuvio 8).

$$\underbrace{\text{AAAAAAAA}}_{8\text{kpl}} \underbrace{\text{VVVVVVVV}}_{8\text{kpl}} \underbrace{\text{DDDD}}_{4\text{kpl}} \underbrace{\text{FFF}}_{3\text{kpl}} \underbrace{\text{DDDDDDDDDDDDDDDDDDDD}}_{16\text{kpl}} \xrightarrow{\text{koodaus}} 8A8V4D3F16D$$

Kuvio 8. RLE koodin toiminta

Esimerkin tapauksessa merkkijono on RLE koodattuna "8A8V4D3F16D".

Moniväristen kuvien tullessa käyttöön Steve Wilhite kehitti CompuServelle GIF formaatin (engl. Graphics Interchange Format), joka pystyi käsittelemään useampia värejä (CompuServe Incorporated 1987, s. 3). GIF:stä tulikin 80-luvulla suosittu tehokkuutensa ansiosta (O’Leary 2013). GIF:n tehokkuus pohjautui sen käyttämään LZW- pakkausmenetelmään.

3.2.3 LZW

Welch (1984) esitti artikkelissaan *A technique for high-performance data compression* häviättömiin pakkausmenetelmiin kuuluvan Lempel-Ziv-Welch-pakkausmenetelmän (LWZ). Siinä toistuvat rakenteet koodataan tunnetun koodiston avulla, ja vastaanottaja pystyy muodostamaan purkukoodin saman tunnetun koodiston avulla. Algoritmin kehitti Terry Welch Abraham Lempelin ja Jacob Zivin LZ78 algoritmin pohjalta, ja se on nykyäänkin yleisesti käytössä.

Pakkaus perustuu siihen, että on olemassa perusavaimet, joilla viesti saadaan pakattua ja purettua. Perusavaimien kombinaatioista saadaan lisää avaimia, ja pakkaaminen tapahtuu määrittelemällä pidempiä avaimia ja käyttämällä toistuvia rakenteissa aiemmin määriteltyjä koodeja (Kuvio 9).

Syöte	a	b	a	b	c	b	a	b	a	b	a	a	a	a	a	a	a
Lähtökoodi	1	2	4	3	5	8		1	10	11							
Koodistoon lisättävä rivi	5		7			9			11								
	4		6			8			10		12						

Kuvio 9. LZW koodaus (mukaellen Welch 1984, s. 16)

Purkaminen menee päinvastoin, eli koodisto rakennetaan avaimen perusteella. Uusia koodeja luodaan jo saadun tiedon perusteella (Kuvio 10).

LZW-koodauksesta on myöhemmin kehitetty erilaisia versioita joissa toimintaa on koetettu tehostaa. Tällainen on muun muassa S-LZW- algoritmi (Sadler ja Martonosi 2006, s. 269-

Syöte	1	2	4	3	5	8	1	10	11
	v	v	v	v	v	v	v	v	v
	a	b	1b	c	2a	5b	a	1a	10a
			v		v	v		v	v
			a		b	2a		a	1a
						v			v
Purettu koodi	a	b	ab	c	ba	bab	a	aa	aaa
Koodistoon lisättävä rivi	<u>4</u>		<u>6</u>		<u>8</u>		<u>10</u>		
		<u>5</u>		<u>7</u>		<u>9</u>		<u>11</u>	

Kuvio 10. LZW dekoodaus (mukaellen Welch 1984, s. 16)

270). S-LZW on tarkoitettu akkukäyttöisten laitteiden tiedonsiirtoon. Se perustuu tekniikkaan jossa mittaustietoa kerätään määrätynkokoinen lohko joka koodataan ja lähetetään. Tällä tavalla saadaan suuria säästöjä lähetetyn tiedon suhteen, mutta se ei ole aivan reaaliaikainen, koska tieto pitää tallentaa puskuriin ennen pakkausta.

3.2.4 Lineaarinen paloittain määritelty funktio

Lineaarinen paloittain määritelty funktio (engl. piecewise linear function) määritellään monessa osassa erillisissä alifunktioissa. Tällä tavoin voidaan määritellä eri funktiot toteuttamaan tiettyjä muuttujan arvoja.

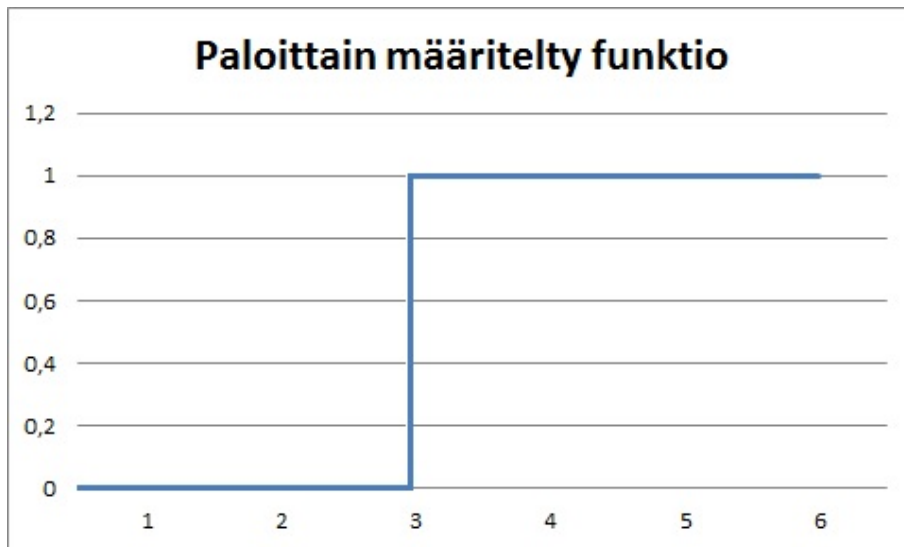
Esimerkiksi voimme määritellä askelvasteen seuraavasti:

$$\chi_A(x) = \begin{cases} 1, & x \geq 3 \\ 0, & x < 3. \end{cases}$$

Tämä tarkoittaa, että kun x:n arvo on kolme tai suurempi, niin funktion arvo on yksi, ja kun x:n arvo on pienempi kuin kolme, saa funktio arvoksi nolla. Sama kuvaajana esitettynä (Kuvio 11).

Tämä on varsin kätevä tapa saada määriteltyä koko määrittelyväli yhdeksi funktioyhmäksi.

Tekniikka on tutkielman kannalta tärkeä koska ennustamisen menetelmät käyttävät paloitt-



Kuvio 11. Paloittain määritellyn funktion kuvaaja

taista ajattelua, eli tietyllä aikavälillä toiminta on erilaista verrattuna edellisiin.

3.2.5 Liukuva ikkuna

Liukuva ikkuna (engl. sliding window) on datajoukosta otettu tarkastelujoukko. Tekniikka on käytössä eri variaatioineen erilaisissa sovelluksissa yksinkertaisuutensa vuoksi. Ajatuksena liukuvaa ikkunaa voidaan pitää eräänlaisena FIFO-rekisterinä (engl. First In First Out). Siinä rekisteriin tallennetaan tietoa toisesta päästä ja pulautetaan aina vanhin tieto ulos toisesta päästä (Kuvio 12). Ikkunan sisällä oleville tiedoille voidaan sitten tehdä halutut toimenpiteet.

Aika		1	2	3	4	5	6	...	n-2	n-1	n
Datavirta	...	a	b	c	d	e	f	...	n-2	n-1	n
...											
Ikkuna ajanhetkellä 3		a	b	c							
Ikkuna ajanhetkellä 4			b	c	d						
Ikkuna ajanhetkellä 5				c	d	e					
...											
Ikkuna ajanhetkellä n									n-2	n-1	n

Kuvio 12. Liukuva ikkuna

Liukuva ikkuna voi olla myös dynaaminen, jolloin ikkunan kokoa muutetaan suorituksen aikana. Tällöin ikkunan kokoa kasvatetaan kun ei poisteta ikkunasta vanhinta tietoa tai pienennetään poistamalla vanhinta tietoa.

4 Signaalin tunnistus

Signaalin tunnistus on käytössä monessa sovelluksessa. Eräs Gartnerin hypekäyrällekin (Gartner, Inc. 2016) päässyt asia on puheen kääntäminen ja puheohjaukset. Nämä ovat olleet jo muutaman vuoden ajan kuluttajien laitteissa varsin toimivina, kuten Applen Siri ja Androidin Googlen puheohjaus. Puheen tunnistaminen on eräs niistä asioita joissa signaalintunnistamista käytetään, ja vaikka se on ollut kiinnostuksen kohteena jo pitkään, alkaa se vasta nykyään olla käyttäjäystävällisellä tasolla. Puheen tunnistaminen kuitenkin on hyvä esimerkki miten signaalintunnistus tapahtuu, signaali sovitetaan johonkin malliin ja pystytään mallin avulla tunnistamaan (Picone 1993, s. 1215).

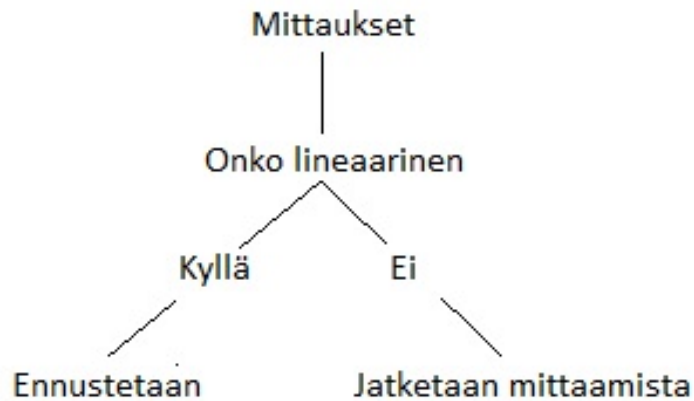
4.1 Koneoppiminen signaalin tunnistuksessa

Koneoppiminen (engl. machine learning) on eräs tekoälyn (engl. artificial intelligence) alue jolla pyritään etsimään erilaisista tietomääristä haluttuja ominaisuuksia. Koneoppiminen voi olla täysin opetettua tai sitten kone itse oppii käytettävissä olevasta tiedosta. Opetukseen käytettävän tiedon valinta voi tosin olla haasteellista tiedon runsauden vuoksi ja sen mihin päämäärään halutaan oppimisella päästä. Oppiminen voidaankin luokitella kahteen gategoriaan: mitä ominaisuuksia oppimiseen liitetään, ja miten ominaisuuksia käytetään (Blum ja Langley 1997, s. 245-246). Karkealla jaolla koneoppimisen tutkimussuunnat voidaan jaotella kolmeen osaan: tehtäväpainotteiseen, ihmistä mallintavaan ja teoreettiseen mallinnukseen. Tehtäväpainotteinen tunnetaan myös teknisenä lähestymistapana (engl. engineering approach). (Michalski, Carbonell ja Mitchell 2013, s. 3-4) Tässä tutkielmassa lähestytään koneoppimista tehtäväpainotteisesti.

Koneoppimisessa käytettyjä tekniikoita on monia. Yleistettynä voi sanoa että koneoppiminen on valmiiden tai uusien mallien etsimistä datajoukosta. Mallien etsimiseen on useita erilaisia keinoja joista useat liittyvät tilastollisiin menetelmiin.

Simon Herbert (Michalski, Carbonell ja Mitchell 2013, s. 28) määrittelee oppimisen tapahtumaksi jossa sama suoritus voidaan tehdä nopeammin seuraavalla kerralla. Opitaan siis tekemään sama asia tehokkaammin. Signaalilla, joka on kiinni samassa prosessissa, tapahtu-

mat ovat usein toistoja edellisistä tapahtumista. Voisi olettaa koneoppiminen soveltuvan hyvin tällaisille toistuville signaaleille. Koneoppimisen tekniikoista päätöksentekopuu (engl. decision tree learning) on yksinkertainen tapa tehdä luokittelu asioille, jotka voidaan jakaa luokkiin ennalta määrätyn ominaisuuden perusteella (Mitchell 1997, s. 52-53). Päätöksentekopuussa opetetaan opetusjoukolla (engl. training set) ja päätökset tehdään luokitusääntöjen (engl. classification rule) pohjalta (Quinlan 1986, s. 86). Tutkielmassa tätä tekniikkaa voidaan käyttää siten, että mittaustiedot toimivat opetusmateriaalina ja päätöksentekopuu määrittelee signaalimallin jota lähdetään ennustamaan (Kuvio 13). Erilaisia signaalimalleja voidaan lisätä puuhun luokitusääntöjä lisäämällä. Järjestelmä oppii signaalin aina reaaliaikaisesti.



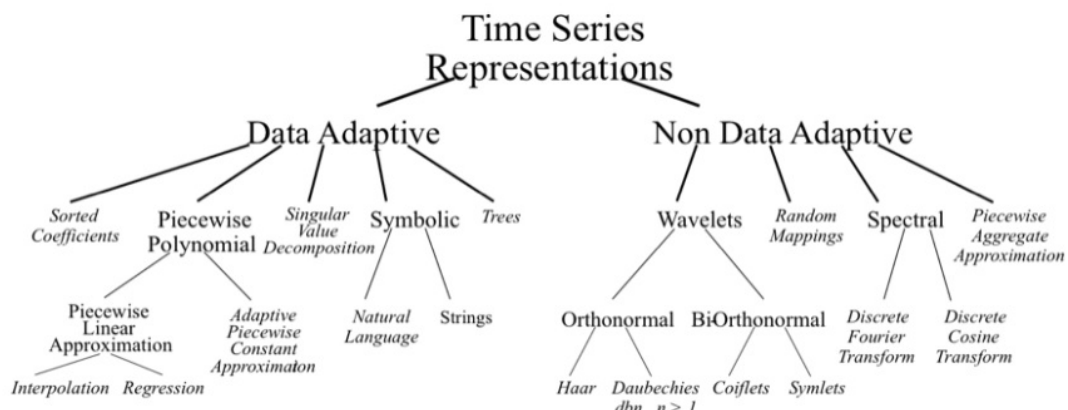
Kuvio 13. Tutkielmassa käytetty luokittelu päätöksentekopuussa

4.2 Aikasarja

Brockwell ja Davis (2002, s. 1-2) määrittelevät aikasarjan peräkkäisistä havainnoista tallennetuksi havaintosarjaksi. He jakavat aikasarjat kahteen ryhmään, diskreettiin- ja jatkuvaan aikasarjaan. Diskreetissä aikasarjassa tallennusväli on kokonaislukuindeksillä esitettävissä ja jatkuvassa tallennetaan tietoa jonkin aikaikkunan yli. Tyypillisiä aikasarjoja on kaikki elämään liittyvät asiat, jotka ovat kellonajan perusteella sijoitettavissa tietyllä aikavälillä.

Mittaustietoa on aikasarjoissa paljon. Yksi suurimmista isojen datamäären tuomista ongelmista on datan esittäminen. Esitys on osa datan tehokkuutta ja hyvä esitysmuoto auttaa te-

hokkaiden ratkaisujen löytämiseen. (Keogh ym. 2001, s. 289) Aikasarjojen esitykseen on kehitetty erilaisia menetelmiä, kuten Fourierin muunnos, symbolinen esitys ja paloittainen lineaarinen esitys (engl. Piecewise Linear Representation (PLR)). Lin ym. (2012, s. 7-8) ovat koonneet kirjallisuudesta erilaisia tekniikoita näiden esittelyyn ja menetelmiä on monia (Kuvio 14). Menetelmä riippuu aineistosta ja siitä, mitä sen avulla halutaan saada selville.



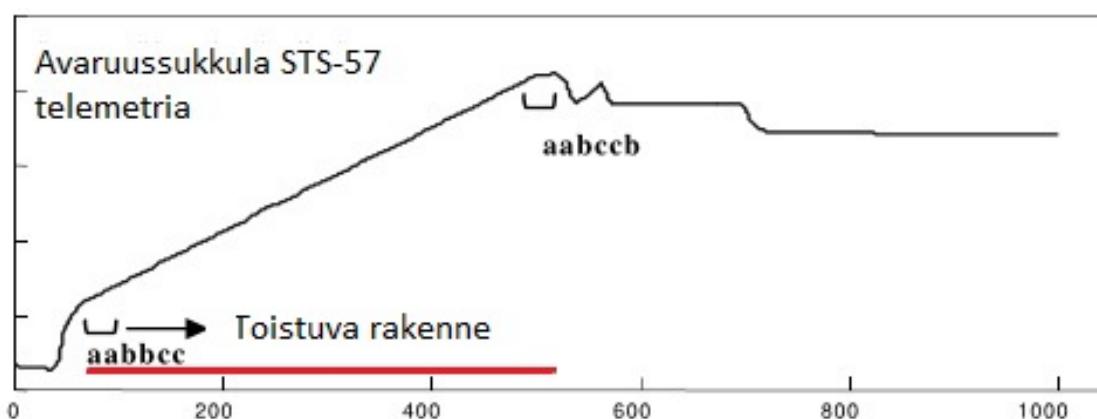
Kuvio 14. Kirjallisuudessa esiintyvät aikasarjaesitykset (Lin ym. 2012, s. 8)

Tässä tutkielmassa keskitytään paloittaisen lineaarisen esityksen käsittelyyn, joka on suosittu tapa käsitellä aikasarjoja. Tälle on periaattellisesti kolmenlaista ratkaisumallia, liukuvan ikkunan (engl. sliding window), alhaalta-ylös (engl. bottom-up) ja ylhäältä-alas (engl. top-down) ratkaisumallit. Kaikissa näissä aikasarjaa lähdetään tutkimaan järjestelmällisesti ja jonkin kriittiseksi asetellun arvon jälkeen määritellään uusi lineaarinen jakso. (Keogh ym. 2001, s. 290-292) Näistä kolmesta liukuvan ikkunan menetelmä on ainoa joka soveltuu reaaliaikaiseen käyttöön ja siksi valitaan tutkielman menetelmäksi (Taulukko 1). Taulukosta voidaan lisäksi nähdä, että liukuvan ikkunan menetelmä on kompleksisuudeltaan helppo ja säästää laskentaresursseja.

Taulukko 1. Paloittain määritellyt aikasarja algoritmit (mukaillen Keogh ym. 2001, s. 292).

Algoritmi	Online	Kompleksisuus
Ylhäältä-Alas	Ei	$O(n^2K)$
Alhaalta-Ylös	Ei	$O(Ln)$
Liukuva ikkuna	Kyllä	$O(Ln)$

Aikasarjojen analysointia käytetään muun muassa maanvärähtelyn mittaamisessa sekä avaruuden tutkimisessa. Kuten edellisistä esimerkeistä voi olettaa, tallennetut tietomäärät kasvavat tämäntapaisissa aiheissa hyvin suuriksi. Lin ym. (2003, s. 6-7) ovat 2003 on esittäneet RLE:n tapaisen tiedon tiivistämistä, jossa he käyttävät avaruussukkulan telemetria tietoja tiivistyksen kohteena (Kuvio 15). Tekniikka esimerkiksi jakaa Shannonin ajatukset toistuvista rakenteista ja niiden kautta saatavan hyödy tiivistämisestä (Shannon 2001, s. 6-9). Tässä tallennustavassa toistuva rakenne, kuvassa "aabbcc", jätetään tallentamatta kunnes se vaihtuu. Siten säästetään tallennustilaa.



Kuvio 15. Liukuvan ikkunan menetelmällä tiivistys (mukaellen Lin ym. 2003, s. 7)

4.3 Ennustaminen

Ihminen on aina ollut kiinnostunut tulevaisuudesta. Aihepiirin ympärille on muodostunut kaikenlaista liiketoimintaa, niistä osa perustuu tieteeseen ja osa ihmisten hyväuskoisuuteen. Yhteistä näille on Pynnösen (Pynnönen 2005, s. 7-8) mukaan tulevaisuuden ennustamisen luoma etulyöntiasema muihin nähden. Silti tulevaisuuden tietäminen koetaan niin pelottavaksi ettei sitä taitoa kukaan todellisuudessa halua, vaikka siitä haaveillaankin.

Taloustieteessä on jo pitkään oltu kiinnostuneita selvittämään pörssikurssien ja talouden käyttäytymistä tulevaisuudessa. Siihen tarkoitukseen on kehitetty mitä monimutkaisimpia algoritmeja ja ohjelmistoja, jotka ovat niin uskottavia että osa pörssikaupoista käydään ro-bottikaupankäyntinä (engl. High frequency trading (HFT)). Ennustamisen vaikeudesta ker-

too kuitenkin Wall Streetin tapahtumat 2010 (Wikipedia 2016), jolloin osakemarkkinoilla tapahtui Flash Crash:ksi nimetty robottien aiheuttama pörssiromahdus.

Talouden ennustamisessa lähtötiedoille on yhteistä tiedon aikariippuvuus. Oletetaan että muodostuneista aikasarjoista voidaan ennustaa tulevaa käyttäytymistä. Taanilan (Taanila 2011, s. 1) mukaan aikasarjan voidaan jo olettaa sisältävän informaation, jonka perusteella tulevat havainnot voidaan ennustaa. Toisi sanoeni tulevilla tapahtumilla on jonkinlainen riippuvuus-suhde aikaisempiin tapahtumiin. Pynnönen määritteleeekin ennustuksen olevan tämänhetki-seen tietoon perustuva odotus tulevaisuuden tapahtumasta.

Erilaisia ennustamistekniikoita on monia. Brockwell ja Davis (2002, s. 317-330) mainitsevat kolme mallia joilla on onnistuttu ennustamaan tehokkaasti datasarjoista. Nämä ovat ARAR-, Holt-Winter- ja Holt-Winter seasonal- algoritmit. Näiden ennustamistavat ovat erilaisia ja onnistumistodennäköisyys samoin. Brockwell ja Davis ohjeistavatkin ennustusmallin valin-nassa vertaamaan ennustuksen ja historiatiedon virheitä parhaan mallin löytämiseksi.

Tässä tutkielmassa lähestytään ennustamista kuitenkin huomattavasti kevyemmin menetel-min kuin aiemmin mainitut. Näitä yksinkertaisempia ovat esimerkiksi liukuvan keskiarvon menetelmä ja lineaarinen regressioanalyysi. Seuraavaksi tarkastellaan tarkemmin niitä ja nii-den antamia mahdollisuuksia.

4.3.1 Liukuva keskiarvo

Liukuva keskiarvo on kahden pisteen välisten arvojen väliltä laskettu keskiarvo. Talouden analysoinnissa käytettyjä keskiarvoja on periaatteessa kahdenlaisia, yksinkertainen liukuva keskiarvo (SMA) (engl. Simple Moving Average) (Investopedia, LLC. 2017b) ja painotettu liukuva keskiarvo. Yksinkertainen liukuva keskiarvo lasketaan yksinkertaisesti alku- ja pää-tepisteen väliltä keskiarvona ja painotetussa liukuvassa keskiarvossa lasketaan painotetaan jollakin perusteella, esimerkiksi eksponentiaalisesti (EMA) (engl. Exponential Moving Ave-rage) (Investopedia, LLC. 2017a) tai muuttujakohtaisesti painottaen (WMA) (engl. Weighted Moving Average) (Investopedia, LLC. 2017c).

4.3.2 Lineaarinen regressioanalyysi

Regressioanalyysi on Mellinin mukaan ehkä eniten sovellettu menetelmä riippuvuussuhteiden analysointiin. Regressioanalyysi voidaan jakaa lineaarisiin ja epälineaarisiin regressiomalleihin. (Mellin 2007b, s. 4-7) Näistä lineaarinen on yksinkertaisempi. Tutkielmassa hyödynnetään sen tekniikkaa.

Lineaarinen regressioanalyysi on tilastollinen menetelmä jonka avulla tutkitaan selittävien muuttujien vaikutusta selitettävään muuttujaan (Kvantitatiivisten menetelmien tietovaranto 2016). Tekniikalla voidaan ennustaa toisistaan riippuvien muuttujien käytöstä aineistosta saatavan mallin mukaan. Riippuvuudesta voidaan piirtää regressiosuora joka kuvastaa muuttujien riippuvuuden voimakkuuden (4.1) (Kvantitatiivisten menetelmien tietovaranto 2016):

$$Y = a + bX \quad (4.1)$$

missä

Y	on	selitettävän muuttujan arvo
b	on	regressiokerroin
a	on	vakio-osa

Yhtälön muuttujat saadaan ratkaistua muun muassa myöhemmin esiteltävän pienimmän neliosumman menetelmällä.

5 Ennustava protokolla

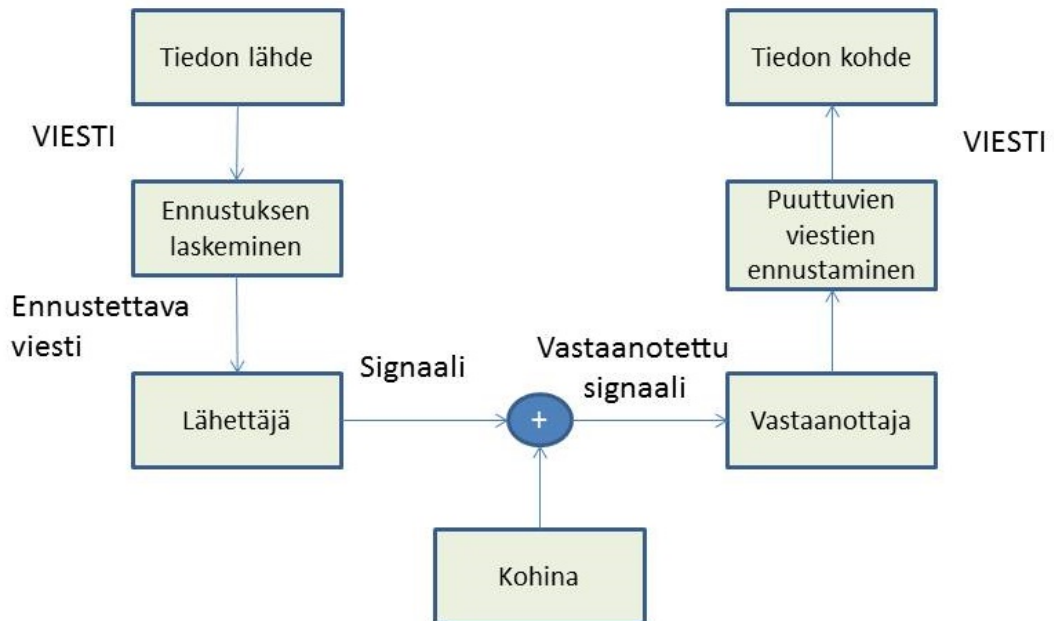
Tässä tutkielmassa selvitetään onko mahdollista säästää resursseja siirtämällä tietoa tunnistavalla protokollalla. Yleisesti käytettävässä aikaperusteisessa tiedonsiirrossa tiedonsiirto on syklistä ja siinä siirretään paljon samaa tietoa. Tämä on syy aihepiirin on kiinnostavuuteen. Voidaanko käyttäytyminen ennustaa historian perustella ja siten säästää resursseja?

5.1 Teoria

Shannon on esitellyt yleisen tiedonsiirtojärjestelmän (Kuvio 2). Siinä järjestelmä lähettää tietoa ja vastaanottaja vastaanottaa sen siirtoprosessissa tapahtuneiden häiriöiden kanssa.

Tässä yhteydessä voidaan soveltaa määrähäiriöteoriaa (engl. rate-distortion theory) (Shannon 1959, s. 328-; Ortega ja Ramchandran 1998, s. 23-24) eli teoriaa siitä, miten pienellä määrällä tietoa voidaan vastaanottopäässä palauttaa lähtötilanne sallitulla häiriöllä. Määrähäiriöteoria johtaa päätelmään, että kaikki turha tieto kannattaa poistaa, mikäli se on mahdollista. Jos osa tiedosta voidaan ennustaa, vähentää se siirrettävän tiedon tarvetta. Tämä voidaan myös esittää siten, että kaikki ennustettavissa oleva tieto ei kuormita tiedonsiirtoa, ja ennustuksella tiedonsiirtokapasiteettia voidaan kasvattaa suorassa suhteessa ennustusmallin toimivuuteen.

Mitä tämä tarkoittaa käytännössä onkin monialaisempi kysymys. Ennustamisen avulla mitaussovellutuksessa voidaan pienentää näytteenottoväliä ilman että järjestelmä kuormittuu. Samalla säästetään siirtotien energiantarvetta ja tiedon tallentamiseen tarvittavaa kapasiteettia. Järjestelmä tulisi olemaan kuvan 16 mukainen.



Kuvio 16. Ennustava kommunikointijärjestelmä

Protokollan avulla signaali pyritään tunnistamaan ennen lähetystä ja lähettämään pelkästään se määrä tietoa, jolla tieto saadaan mallinnettua vastaanottopäässä.

Protokollasta saatavan hyödyn selventämiseksi käydään läpi kaksi esimerkkiä. Ensimmäinen järjestelmä on nykyaikainen aikaperusteinen järjestelmä ja toinen esimerkki kertoo ennustavaa protokollaa käyttävästä järjestelmästä.

Esimerkissä seurataan yksittäistä lämpötilatietoa kasvihuoneessa. Kasvihuoneen lämpötila on tärkeä kasvien hyvinvoinnille ja sitä mitataan että voidaan seurata olosuhteita. Lämpötila luetaan reaaliaikana, jolloin yhden mittauksen vaatima tallennustarve on 4 tavua. Lämpötila on mahdollista saada lämpötilaa mittaavasta järjestelmästä kerran minuutissa. Tieto tallennetaan pääjärjestelmän tietokantaan jonne se siirretään langattomasti matkapuhelinverkon läpi. Järjestelmä suorittaa tiedosta hälytyksen jos lämpötila nousee tai laskee asetettujen rajojen ulkopuolella. Hälytyksestä voidaan suorittaa haluttuja toimenpiteitä, kuten esimerkiksi soittaa huoltoyhtiöön.

Ensimmäinen järjestelmä mittaa tietoa perinteisesti. Järjestelmän suunnittelija on määritellyt

tilaajan kanssa yhden tunnin tallennusvälistä. Tämä määritellään tunniksi siksi, että lämpötila ei normaalisti muutu nopeasti, ja lähettävän laitteen akku kestää pidempään kun lähetystä ei suoriteta usein. Lisäksi tallennusvälin määrittelyyn vaikuttaa tietoa keräävän järjestelmän kuormittuminen, mikäli kaikki siihen liitettävät laitteet tuottavat tietoa lyhyellä aikasyklillä.

Toinen järjestelmä käyttää ennustavaa menetelmää. Lämpötilaa tallennetaan silloin kun se muuttu merkittävästi, ja koska se luonteensa vuoksi muuttu niin hitaasti, ei tallennusta tarvitse muuten rajoittaa.

Kumpikin järjestelmä toimii hyvin, mutta toinen järjestelmä kertoo lämpötilan muutoksesta minuutin tarkkuudella kun ensimmäinen reagoi muutokseen jopa tunnin viiveellä. Koska lämpötila muuttuu hitaasti, kestää toisen järjestelmän akku paljon pidempään, koska lähetystä ei tarvitse tehdä usein. Lisäksi huolto ehtii estämään mahdolliset pakkasvauriot, koska lämmitysjärjestelmässä esiintyvät häiriöt huomataan heti eikä tunnin viiveellä.

5.2 Ennustaminen

Miten signaali voitaisiin sitten ennustaa? Usea tilasignaali on luonteeltaan sellainen että staattisia tiloja on paljon tai muutos on tasainen. Tila seuraa siis lineaarisesti aikaa ja trendille piirtäessä tila näyttää suoraa viivaa.

Tilastotieteen ja taloustieteen puolella ennustamista on käytetty pitkään. Sieltä löytyy apuneuvoja mallintamaan viiva näytteistä. Menetelmä, jolla suora muun muassa saadaan määriteltä, on nimeltään pienimmän neliösumman suora (PNS). Tällä menetelmällä voidaan havaintoaineisto sovittaa lineaariseen malliin ja ratkaisuna saadaan lineaarisen mallin yhtälö. Tässä tutkielmassa tämä menetelmä otetaan käyttöön yksinkertaisuutensa vuoksi ja siksi, että tutkielmassa tarkastellaan vain lineaarisen muutoksen ennustaminen.

PNS menetelmällä saatu yhtälö noudattaa suoran yhtälöä (5.1)(Laitinen ym. 1990, s. 115). Tämän avulla voidaan määrittää osittainen funktio tietylle aikavälille.

$$y = kx + b \tag{5.1}$$

missä

y on pistearvo arvolla x
 k on suoran kulmakerroin
 b on vakio-osa

PNS menetelmällä voidaan ratkaista suoran tarvitsevat muuttujat. Kulmakertoimen ratkaisukaava on (5.2)(Laitinen ym. 1990, s. 115).

$$k = \frac{n \sum x_i y_i - (\sum x_i)(\sum_{k=0}^n y_i)}{n \sum x_i^2 - (\sum x_i)^2} \quad (5.2)$$

missä

k on suoran kulmakerroin
 x ja y on mittausarvo pari

Ja vakion ratkaisukaava on (5.3)(Laitinen ym. 1990, s. 115).

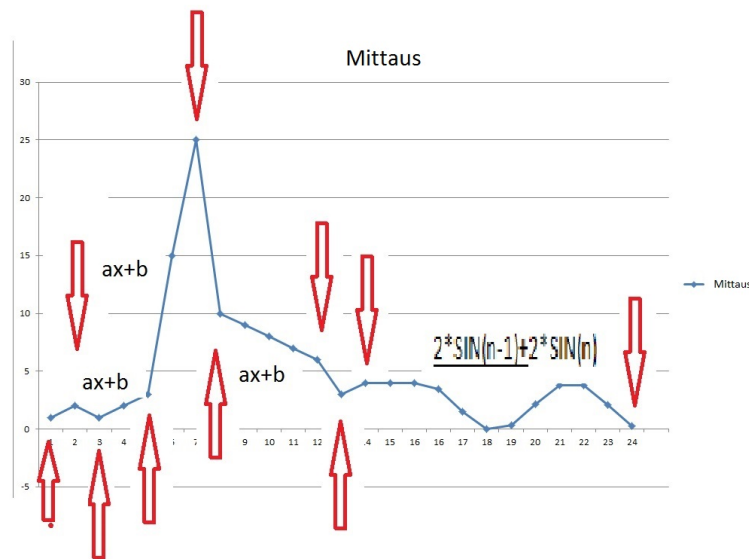
$$b = \frac{(\sum y_i)(\sum x_i^2) - (\sum x_i)(\sum x_i y_i)}{n \sum x_i^2 - (\sum x_i)^2} \quad (5.3)$$

missä

b on vakio-osa
 x ja y on mittausarvo pari

Tutkielmaa varten kehitetään algoritmi joka käyttää hyväksi ennustamista samaan tapaan kuin MPEG kuvan pakkaamisessa ja siirretään tieto kuten videokuva. Ennustamisessa tunnistetaan lineaarisen regressioanalyysin suora ja opetetaan vastaanottaja löytämään se, jolloin siirrettäväksi tiedoksi tarvitaan vain tunnistamisen verran tietoa (Kuvio 17). Tiedonsiirtoprotokollaksi valitaan resurssien säästämisen vuoksi UDP. Tavoite on siis rakentaa eräänlainen paloittain määriteltävä, lineerisen muutoksen tunnistava, reaaliaikainen ja ennustava tiedonsiirtoalgoritmi.

Kuviolla (Kuvio 17) esitetään ennustamisen periaatetta. Kuvion punaiset nuolet osoittavat pisteet joissa signaalin luonne muuttuu ja joista on mahdollista ennustaa signaalin muutos. Konetunnistamisen menetelmillä päätöksentekopuu rakentaa mallia luokittelemalla signaalia



Kuvio 17. Signaalin tunistaminen ja lähetyspisteet

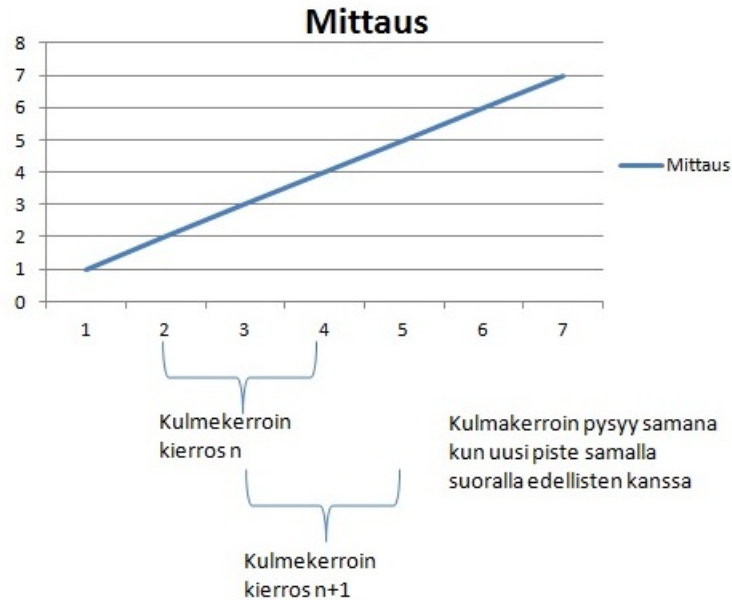
jatkuvasti. Kun mallia varten on tarpeeksi tietoa voidaan siirto keskeyttää, ja vastaanottaja voi saman tiedon perusteella rakentaa mallin, jolla signaali jatkuu.

Idea on, että vastaanottajalle jo lähetetystä tiedosta vastaanottaja saa rakennettua mallin signaalin ennustamiseen. Tämä toimii aiemmin käytyjen ennustamisen keinojen kanssa, kun oletetaan jo tapahtuneen antavan tiedon tulevasta toiminnasta. Tämä tarkoittaa käytännössä sitä, että kun tietoa ei enää saavu, tekee vastaanottaja päätöksen ennustamisesta. Jos tietoa myöhemmin tulee, korjaa vastaanottaja ennustuksen vastaamaan uutta tietoa. Sanomaliikenne on siis järjestysmerkittävä ja vastaanottajan tulee pystyä arvioimaan, onko joku viesti jäänyt välistä mikäli sopivaa mallia ei löydy tietojen perusteella. Puuttuva tietoa voidaan kysyä uudelleen. Käytännössä tietoa tallennetaan ensin ja käytetään jälkikäteen johonkin tarkoitukseen. Tällöin on tärkeää vain saada tieto talteen, ja siihen tällainen myöhemminkin saapunut tieto ehtii hyvin.

5.3 Oletuksia

Jotta tilastollisia menetelmiä voidaan hyödyntää, pitää mittausten olla suhde- tai järjestys asteikolle sijoitettavia, noudattaa normaalijakaumaa, olla homoskedastisia ja eivätkä saa korreloida keskenään (Mellin 2007a, s.5). Nyt ei kaikkia ehtoja voida täyttää, sillä oletettavasti joukko korreloi hyvin voimakkaasti lineaaritapauksessa. Koska mittausjoukkoon ei tehdä muunnoksia, ohitetaan tämä tietoisesti. Näillä rajoitetuilla ehdoilla voimme käyttää pienimmän neliösumman menetelmää (PNS).

Korrelaatiota ei voida käyttää myöskään tarkastamaan mallin yhtäpitävyyttä, koska muuttumattoman signaalin keskihajonta on nolla, eikä korrelaatiota voida tällöin laskea. Lisäksi mittaustulokset ovat absoluuttisia eivätkä sisällä mittaustekniikasta tai ympäristöstä johtuvia virhetekijöitä. Tästä syystä tarkkaillaan pelkästään PNS-menetelmän antamaa kulmakerrointa vertaamaan signaalin suuntaa. Kulmakerroin kolmella mittapisteellä kertoo jos suoran suunta muuttuu ja siten uusi mittaus ei ole lineaarinen edellisten kanssa (Kuvio 18). Tämä yleistys ei oletettavasti toimi kaikilla mittapisteillä, mutta on riittävä tutkielman testiin.



Kuvio 18. Kulmakertoimen laskeminen

5.4 Mallintestaaminen

Malli testattiin käyttämällä Excel-taulukkolaskentaohjelmaa. Mallin testaamisessa rakennettiin Exceliin kaavat ja simulointidata joiden perusteella tarkastettiin mallin toimintai esimerkkitapauksissa. Esimerkkisignaaleina käytettiin kanttiaaltoa, siniaaltoa ja lineaarisesti käyttäytyvää dataa. Mallinnuksen tarkoituksena oli tarkastaa protokollassa käytettävän algoritmin toiminta ennen varsinaisia testejä.

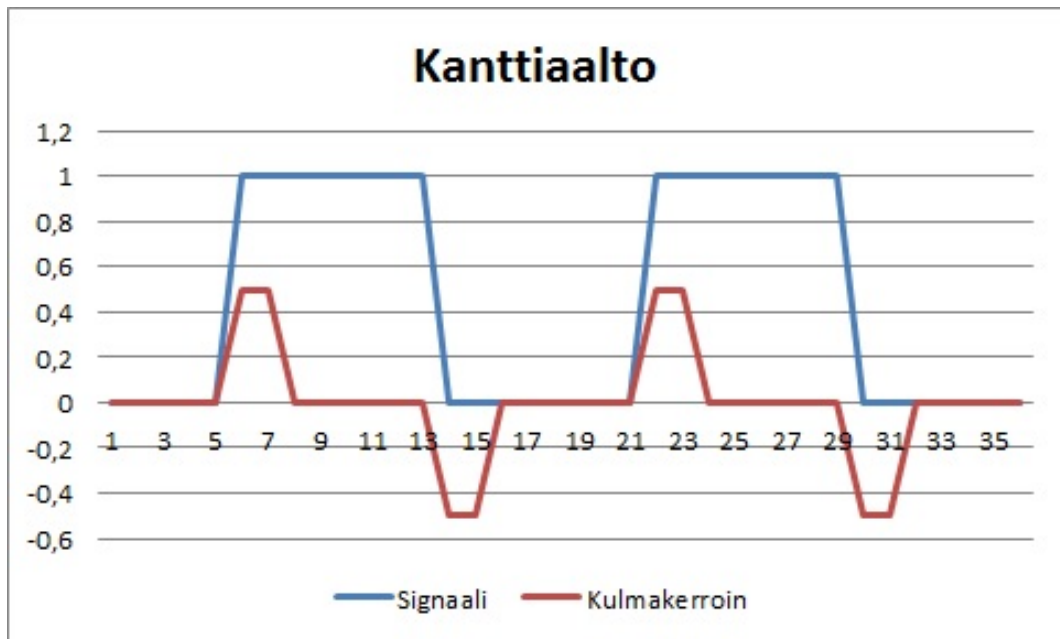
5.4.1 Kanttiaalto

Eräs yleisimmistä signaaleista on Päällä/Pois-tieto. Tämä signaali on yleinen muun muassa häilytyksissä, pintarajoissa ja rajakytkimissä, ja sen ainoa tarkoitus on osoittaa tilan vaihtuminen ja kertoa tila. Signaali on siis hyvin yleinen ja tärkeä tallennettava, ja siksi se soveltuu hyvin testisignaalksi. Signaalilla on myös piirteitä, jotka aiheuttavat signaalinkäsittelyssä tiettyjä ilmiöitä, kuten esimerkiksi Gibbsin ilmiöksi (engl. Gibbs' phenomenon) kutsuttu 9% ylitys Fourier muunnoksessa (MIT 2016). Askelvaste ja impulssivaste ovat kanttiaallon muotoja, joita käytetään paljon järjestelmien testaamisessa. Näistä askelvaste on signaalin nouseva reuna ja impulssivaste pienimmän aikayksikön aikainen signaalin päällä olo (Hsu ja Ranjan 2014, s. 6).

Testaaminen suoritetaan Excelissa valmistamalla aineisto, jossa signaali vaihtuu kahden tilan välillä. Aineistosta lasketaan PNS-menetelmällä kulmakertoimen arvo eteenpäin siirrettävässä aikaikkuna ja tarkastetaan pysyykö kulmakerroin samana signaalin lineaarisilla osuksilla.

Kyseisen mallin testaamiseen kanttiaallosta löydetään tilanne (Kuvio 19) jossa signaali ei enää muutu, eli kulmakerroin pysyy vakiona, jolloin se voidaan mallintaa lineaarisen riippuvuuden keinoja käyttäen.

Kuten kuvaajasta (Kuvio 19) nähdään, seuraa kulmakerroin signaalin muutosta ja malli toimii kanttiaallolla ja voidaan käyttää mallintamiseen.



Kuvio 19. Kanttiaallon mallinnus

5.4.2 Sinitesti

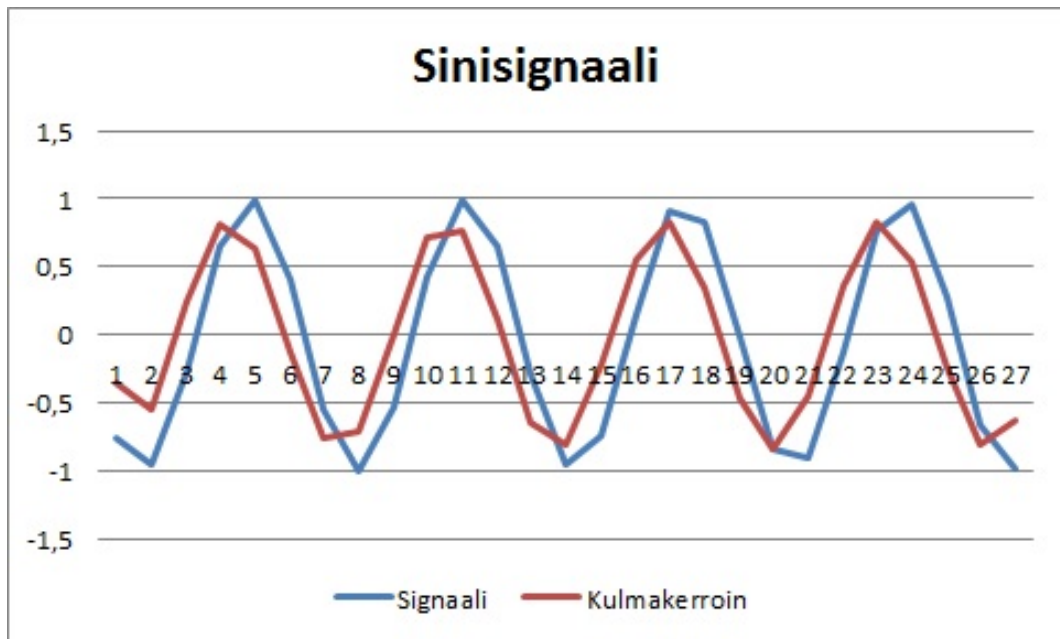
Sinisignaali on tyypillinen vaihtuva signaali. Sen kulmakerroin vaihtuu jatkuvasti ja siksi se on soveltuva mallin testisignaalkiksi (Hsu ja Ranjan 2014, s. 6).

Testaaminen suoritetaan Excelissa valmistamalla aineisto, jossa signaaliin on tallennettu sinin arvoja tietyille ajanhetkille. Aineistosta lasketaan PNS-menetelmällä kulmakertoimen arvo ja tarkastetaan onko kulmakerroin sama liukuvan ikkunan menetelmällä lukujonoa edessä.

Kuten kuvaajasta (Kuvio 20) nähdään, sinisignaali ei korreloi edellisen tiedon kanssa eli kulmakerroin vaihtuu jatkuvasti. Malli siis toimii, koska tarkoitus on tunnistaa lineaarinen riippuvuus, eikä sitä sinisignaalissa ole.

5.4.3 Linearisesti muuttuva signaali

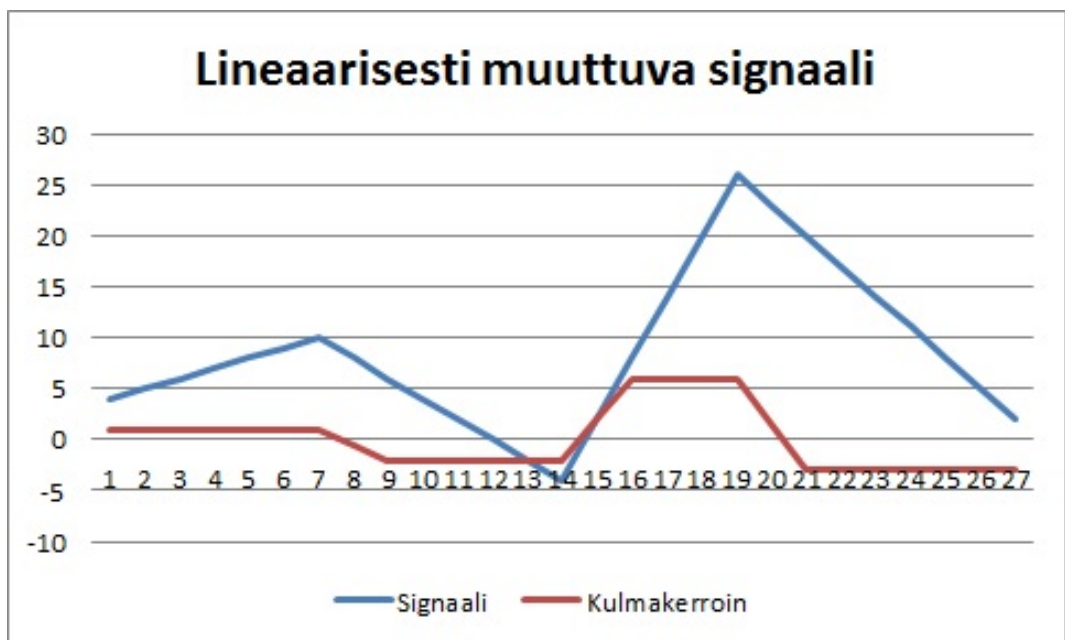
Tutkielman tarkoituksena on testata lineaarinen signaalin ennustavaa menetelmää. Tästä syystä testaaminen suoritetaan käsin rakennetulla, lineaarista muutosta sisältävällä tiedolla.



Kuvio 20. Sinisignaalin mallinnus

Testaaminen suoritetaan Excelissa valmistamalla aineisto, jossa signaaliin on tallennettu lineaarisesti muuttuvia arvoja tietyille ajanhetkille. Aineistosta lasketaan PNS-menetelmällä kulmakertoimen arvo ja tarkastetaan, onko kulmakerroin sama liukuvan ikkunan menetelmällä lukujonoa edetessä.

Kuten kuvaajasta (Kuvio 21) nähdään, signaalista lasketusta kulmakertoisemesta voidaan tunnistaa lineaariset osuudet. Malli toteuttaa vaatimuksen lineaarisen muutoksen tunnistamisessa hyvin.



Kuvio 21. Lineaarisesti muuttuvan signaalin mallinnus

6 Testijärjestelyt

6.1 Testijärjestelmä

Testijärjestelmän on tarkoitus kuvata pienellä laskentakapasiteetilla olevan laitteen mahdollisuuksia lähettää tietoa. Ajatuksena on, että laite vastaa mahdollisuuksien mukaan markkinoilla olevia ethernet liitynnäisiä mittalaitteita. Siksi järjestelmän ja laitteiden osalta valintakriteereiksi muotoutuivat seuraavat:

1. helppo saatavuus
2. helposti opeteltavissa oleva ohjelmointikieli
3. riittävän heikko laskentakyky lähettävässä laitteessa
4. riittävät ohjelmakirjastot
5. hyvät liityntämahdollisuudet
6. tiedostoon tallennusmahdollisuus ja
7. tiedoston lukumahdollisuus.

Näillä kriteereillä testijärjestelmäksi valitaan Arduino mikrokontrolleri ja Pythonilla toteutettu tiedonkeruupalvelin (Kuvio 22). Mittaustiedon siirto toteutetaan sisäverkon kautta, ettei ulkopuoliset häiriöt haittaa testaamista. Lähettävän mikrokontrollerin ohjelmointitapahtuu Arduinon omalla C++:n kaltaiselle Arduno 1.0.5-r2 ohjelmointikielellä ja palvelimen ohjelma toteutetaan Python 3.5.2 ohjelmointikielellä. Laitteisto valintaa vaikuttavat kirjoittajan omat mieltymykset ja aikaisemmat kokemukset laitteista ja ohjelmointikielistä.

6.2 Testimateriaali

Testimateriaalina käytetään kokoelmaa erilaisia mittaustietoja. Testauksessa mallinnetaan kanttiaalto, siniaalto, lineaarisesti muuttava signaali ja ilmatieteenlaitokselta saatava säädataa avaimesta datapalvelusta.

Avoimessa datapalvelussa oleva säädata on kaikkien käytettävissä ja sitä voidaan hyödyntää niin yksityisessä kuin myös kaupallisessa tarkoituksessa. Ilmatieteenlaitoksen säädata on oi-



Kuvio 22. Testijärjestelmä

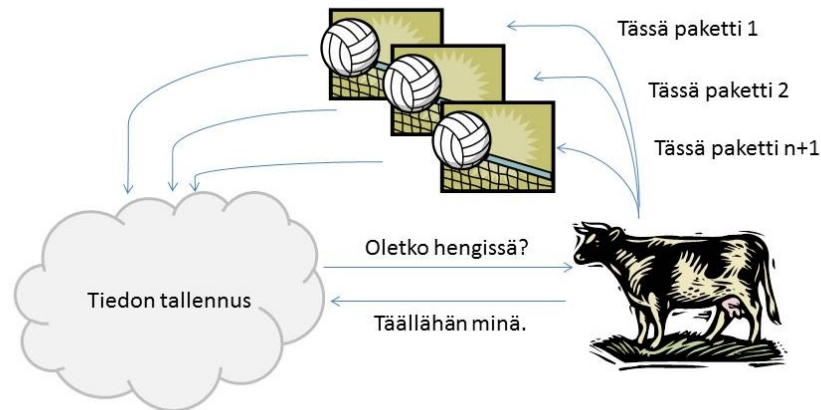
keaa mittaustietoa ja soveltuu siksi hyvin kyseisen tutkielman tarkoituksiin. Itse valmistetulla datalla saadaan vertailua erilaisiin malleihin, jotka ovat tyypillisiä testatessa järjestelmän kykyä eri tilanteissa.

6.3 Ennustava testiprotokolla

Ennustava testiprotokolla käyttää tapahtumapohjaista lähetystä, ettei tarvitse lähettää kuin mallin luomiseen tarvittavat tiedot eteenpäin. Vaikka tämä tekniikka ei ole mainittu yksinkertaiseksi toteuttaa (Decotignie 2005, s. 1103), on se helpoin keino tehdä ennustava protokolla. Siirtoprotokollaksi valikoitui yhteydetön protokolla UDP (engl. User Datagram Protocol), jolla lähetetään mittaustieto eteenpäin (Kuvio 23). UDP on valittu koska se säästää sanomaliikennettä eikä siten kuormita verkkoa ylimääräisesti. UDP:n huonopuoli on sisäänrakennetun virheentarkastuksen puuttuminen, mutta se ei ollut tämän tutkimuksen kannalta oleellista, ja valinta on tietoinen. Virheentarkastus on mahdollista rakentaa käyttämällä esimerkiksi CRC:tä (engl. Cyclic redundancy check) tarkastamaan oikeellisuus. Tämäntapainen virheentarkastus on käytössä muun muassa Modbusin sarjaliikenneversiossa.

Protokollan toiminta perustuu siihen että mittaustietoa lähetetään palvelimelle vain tarpeellinen määrä. Palvelin vastaanottaa sanomat ja ennustaa puuttuvat paketit.

Testiohjelmassa luetaan mittaustietoa ja puskuroidaan sitä laskentaa varten kolmen mittauk-



Kuvio 23. Ennustava testi protokolla

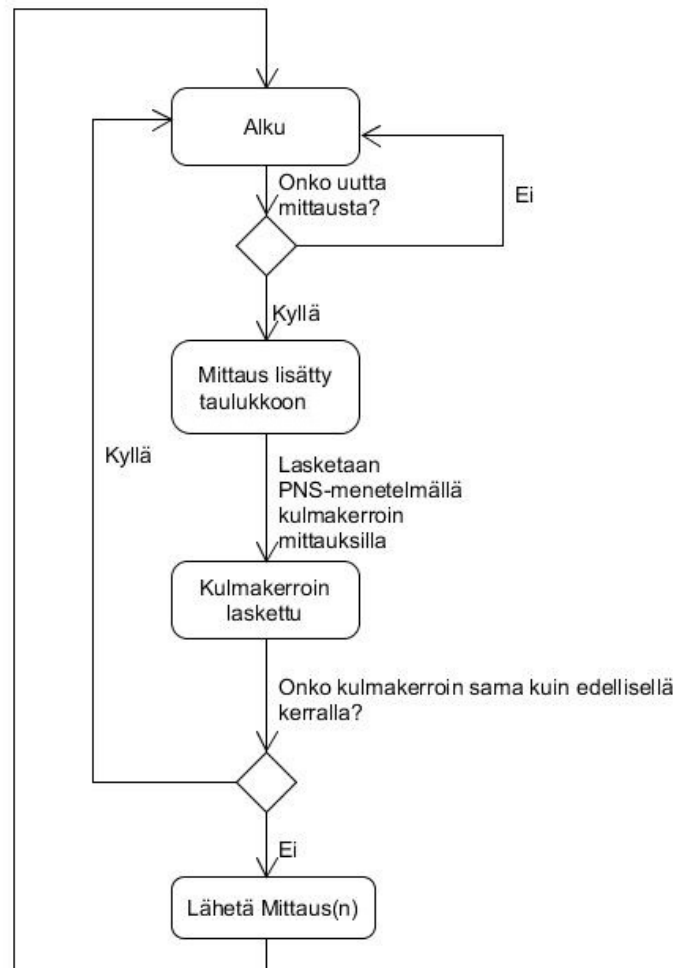
sen verran. Määräksi valikoitui kolme koska kahdesta mittauksesta ei vielä voi nähdä onko mittauksista muodostumassa lineaarista riippuvuutta. Ohjelma on hyvin yksinkertainen eikä ota huomioon mittausepä-tarkkuuden aiheuttamia vaihteluja, mutta tässä testissä sillä ei ole merkitystä. Kolmella tarkastuspisteellä on lisäksi jo mahdollista tunnistaa näytteenotto-välillä vaihtuva pulssitus. Jatkon kannalta mittausepä-tarkkuuden huomioimiseen tulisi lisäksi kiinnittää huomiota, koska sallittu mittausepä-tarkkuus vaikuttaa algoritmin tehokkuuteen parantavasti.

Näytemääräksi on valittu kolme tietoisesti. Patton (1990, s. 169) on maininnut laadullisen ja määrällisen tutkimuksen eron olevan siinä että määrällisessä tutkimuksessa käytettävissä on suuri määrä tietoa, kun taas laadullisessa riittää vähäinenkin tieto kunhan se on tarkoi-tuksenmukaista. Tässä tutkielmassa mittaustieto on myös tarkoituksenmukaista ja tarkkaan kyseiseen kohteeseen eli mittaukseen kohdistuvaa.

6.3.1 Lähetys

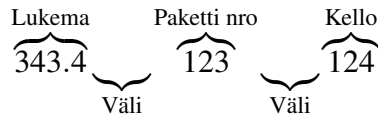
Lähetävässä päässä toiminta on hyvin suoraviivainen (Kuvio 24). Järjestelmä mittaa luke-maa ja taulukoi uuden tuloksen liukuvaan FIFO (engl. First In First Out) taulukkoon. Taulu-kosta lasketaan PNS-menetelmällä lineaarinen riippuvuus ja sovitetaan sitä edellisessä las-kennassa saatuun malliin. Jos malli on sama kuin edellisellä kerralla, mittaustiedon lähetystä

ei tarvitse tehdä, ja jos malli on eri, lähetetään mittausarvo eteenpäin. Mallin sovittaminen on yksinkertaisesti kulmakertoimen tarkkailua. Mikäli kulmakerroin muuttuu, ei uusi lukema ole enää lineaarisesti riippuva edellisistä.



Kuvio 24. Lähettävän ohjelman toiminta

Lähetettävä sanoma sisältää signaalin arvon, pakettitunnisteen juoksevalla numeroinnilla sekä kellonajan (Kuvio 25). Sanoman pakettinumerointi on tärkeä koska se on ainoa informaatio siitä mitkä paketit tulisi ennustaa. Kellonaika on lisävarmistukseksi ja sen perusteella voidaan päätellä puuttuvia osuuksia ja tietoliikenteen viiveitä.



Kuvio 25. Sanomarakenne

6.3.2 Vastaanotto

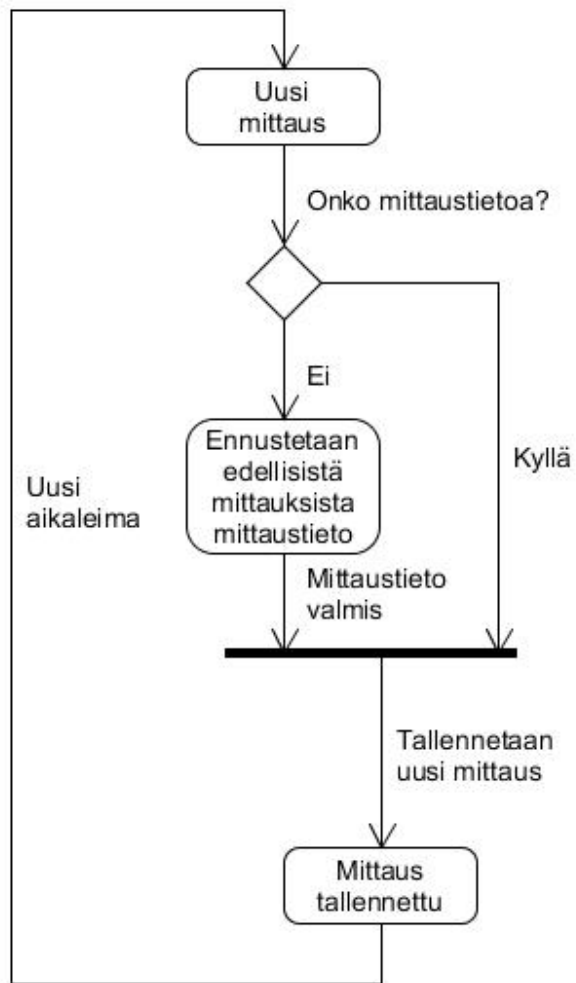
Vastaanottavassa päässä mallinnus tehdään sitten nurinpäin (Kuvio 26). Lähetetyistä pisteistä voidaan laskea sama kulmakerroin, jonka perusteella lähetävä laite on lopettanut tietojen lähettämisen ennustettavana signaalina. Tieto käyttäytymisestä siirtyy toisin sanoen tiedon mukana ennustamisen periaatteiden mukaan, ilman että tarvitsee erikseen siirtää tietoa ennustuksesta.

Vastaanottaja tietää puuttuvat paketit pakettinumeroiden perusteella ja voi suorittaa ennustuksen. Esimerkiksi kun lähetävä pää on lähettänyt 3 ykköstä pakettinumeroilla 120, 121, ja 122, tietää vastaanottaja että paketista 123 eteenpäin tulee ykkösiä kulmakertoimella 0, kunnes tulee lisätietoa.

Tutkielmassa vastaussanomien purkaminen on toteutettu Excel-tilukkolaskenta-ohjelmaa hyväksikäyttäen (Kuvio 27). Excelissä on taulukoitu vastaussanomien pakettinumeron perusteella ja ennustettu puuttuvat paketit algoritmiä hyödyntäen. Exceliin on rakennettu sama algoritmi kuin lähettävään päähän, eli lasketaan PNS-menetelmän avulla suoran pistettä pakettinumeroiden perusteella käyttäen hyväksi kaavoja (4.1), (5.3), (5.2) ja (5.1). Laskenta tapahtuu siis samoilla arvoilla kuin lähettävässä päässä. Lähetäjä on lähettänyt kaikki tiedot joiden perusteella tekee päätöksen ennustamisesta, ja ne ovat vastaanottajalla käytettävissä.

Ennustamiseen käytetään kolmea mittausta. Kuvan Excell taulukosta nähdään mittauksista saatavat arvot, kolmen edellisen mittauksen antamat suoran vakio ja kulmakerroin joiden perusteella voidaan ennustaa aina seuraava lukema. Tätä ennustettua lukemaa käytetään aina seuraavan ennustuksen laskennassa jolloin samalla tekniikalla voidaan ennustaa rajattomasti eteenpäin.

Ennustaminen tapahtuu taulukossa siten että pakettinumeroiden oletetaan olevan tasavälein lähetettyjä. Tällöin arvo voidaan laskea aina kolmen edellisen mittapisteen avulla edellisen



Kuvio 26. Vastauksen ennustaminen

rivin suoran yhtälöstä kaavalla

$$ennustus_n = 4 * b_{n-1} + a_{n-1} \quad (6.1)$$

Kaavassa (6.1) laskentakierroksen järjestystä osoittaa alaindeksi n. Taulukon muiden sarakkeiden arvot (S_x, S_{xx}, S_y ja S_{yy}) ovat muuttujien summia tai neliösummia kolmelta mittauskierrokselta. Näistä lasketaan kulmakerroin ja vakio.

	Signaali	Paketti	Kello	Ennustus	Sx	Sy	Sxx	Syy	Sxy	a	b
Sanoma: b	5	224	230622		6	20	14	66	28	18,66667	-6
Sanoma: b	6	225	231670		6	15	14	77	32	3	1
Sanoma: b	7	226	232717		6	18	14	110	38	4	1
		227		8	6	21	14	149	44	5	1
		228		9	6	24	14	194	50	6	1
		229		10	6	27	14	245	56	7	1
Sanoma: b	8	230	236772		6	27	14	245	53	10	-0,5
Sanoma: b	6	231	237820		6	24	14	200	44	12	-2
Sanoma: b	4	232	238869		6	18	14	116	32	10	-2

Kuvio 27. Kuvio ennustamistaulukosta Excelissä

7 Testi

Testi suoritetaan käyttämällä lähdeaineistona Ilmatieteenlaitoksen avoimen data-arkiston säädettä sekä rakentamalla käsin testidataa, joilla simuloidaan askelvasteen, sinikäyrän ja lineaarisen muutoksen käyttäytyminen. Näillä mallinuksilla pyritään todentamaan erilaiset tilanteet ja protokollan soveltuvuus erilaisille tapahtumille.

7.1 Verrokkina Modbus/TCP

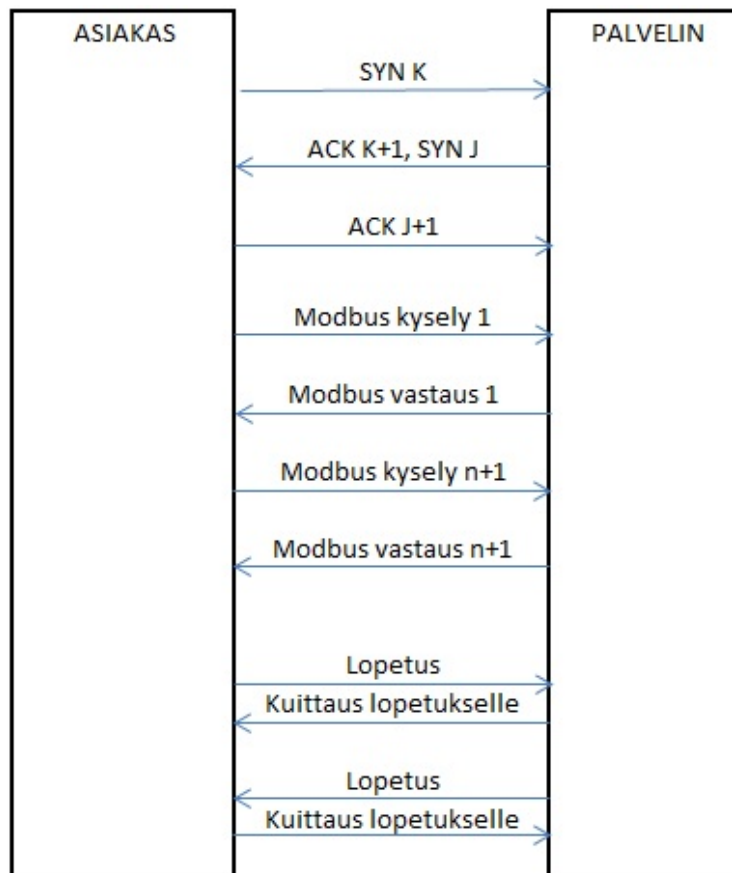
Modbus/TCP käyttää TCP/IP-protokollaa ja sisältää siten sanomaliikenteen virheentarkastuksen. Protokollasta johtuen liikennöintiä tapahtuu kättelyjen vuoksi muutenkin enemmän kuin UDP-protokollaan käyttäen. UDP:llä testiprotokolla vaatii yhden tiedon lähettämiseen vain yhden sanoman, kun taas Modbus kättelyineen, kyselyineen ja vastauksineen huomattavasti enemmän (Kuvio 28).

Kuvassa (Kuvio 29) Modbusin liikennettä ja siitä nähdään protokollan toiminta. Aluksi avataan yhteys ja kun yhteys on valmis, suoritetaan halutun tiedon kysely (Kuvio 28). Tiedon siirron jälkeen yhteys katkaistaan hallitusti. Esimerkissä on käytetty Multiway 10.7 ohjelmaa tekee ilmeisesti virheen lopetuskättelyissä, koska yhteyden lopetus näkyy sanomaliikenteessä vain yhteen suuntaan. Tästä ongelmasta johtuen yhteys jäi kyseisessä ohjelmassa aina päälle.

Modbus/TCP aiheuttaa siis paljon sanomaliikennettä jo pelkän TCP:n takia. Kommunikoinnista on hyvä kuitenkin huomata pollaavan järjestelmän rakenne. Asiakas kysyy, ja siihen vastataan. Tarvitaan siis vähintään kaksi sanomaa tiedon siirtämiseen.

7.2 Ennustava protokolla

Ennustavan protokollan testaamisessa käytetään mittaustietona samaa lähtöaineistoa kuin mallinuksessakin on käytetty. Lähtömateriaali tallennetaan mikrokontrollerin muistikortille ja tiedot luetaan muistikortilta käyttömuistin taulukkoon mikrokontrollerin käynnistyksen yhteydessä. Taulukon arvoja luetaan ja käsitellään 1 sekunnin syklissä. Taulukkoa luetaan



Kuvio 28. Modbus/TCP- kommunikointi (mukaellen Modbus Organization 2006, s. 17)

siten, että lopusta siirrytään takaisin taulukon alkuun, ja ohjelma toistaa toimintaa kierros kierrokselta.

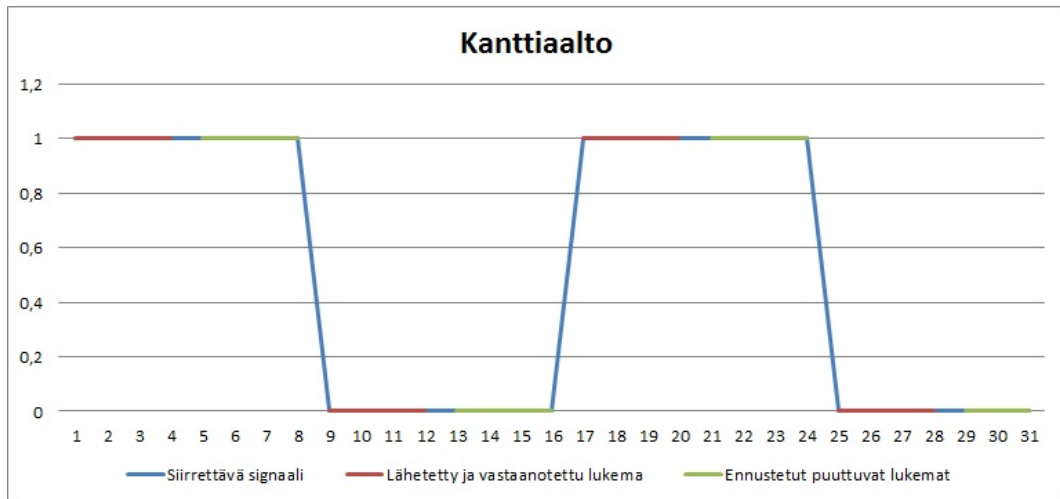
7.2.1 Kanttiaalto

Kanttiaalto tallennettiin Arduinon muistikortille rivikohtaisina arvoina. Arduino luki alkula-
tauksessa muistikortin sisällön taulukkoon ja alkoi suorittaa syklisesti taulukon läpikäyntiä
mallintaen tapahtumaa, jossa luetaan uusi anturin antama arvo. Kuvaajaan (Kuvio 30) on
piirretty siirretty signaali ja ennustetut osiot.

Kuten tuloksista nähdään (Taulukko 2), ennustus toimii hyvin tämän tapaisella signaalilla.
Kuvasta (Kuvio 30) nähdään miten signaalin arvot saatu uudelleenmuodostettua. Sininen
viiva, joka kuvaa alkuperäistä signaalia, on läpi koko kuvaajan päällekkäin muiden viivojen

UTC time	Time	Source IP	Source port	Destination IP	Destination port	Protocol	Packet length
12:48:15.101446	0.000000	192.168.0.13	49575	192.168.0.12	502	TCP	66
12:48:15.141316	0.039870	192.168.0.12	502	192.168.0.13	49575	TCP	66
12:48:15.141398	0.039952	192.168.0.13	49575	192.168.0.12	502	TCP	54
12:48:15.161956	0.060510	192.168.0.13	49575	192.168.0.12	502	Modbus/TCP	66
12:48:15.171875	0.070429	192.168.0.12	502	192.168.0.13	49575	Modbus/TCP	95
12:48:15.189586	0.088140	192.168.0.13	49575	192.168.0.12	502	TCP	54
12:48:16.206250	1.104804	192.168.0.13	49575	192.168.0.12	502	TCP	54
12:48:16.208188	1.106742	192.168.0.12	502	192.168.0.13	49575	TCP	54

Kuvio 29. Modbus sanomaliikennettä



Kuvio 30. Kantiaallon siirtäminen ennustamalla

kanssa. Punainen viiva kertoo mitä tietoja on tarvittu lähettää, ja vihreä viiva kertoo kaikki ennustetut osuudet. Sinisiä osuuksia ei ole näkyvissä kuin epäjatkuvuuskohdissa lähetetyn tiedon lopun, ja ennustetun tiedon alun välillä. Tämä johtuu piirtoteknisistä syistä.

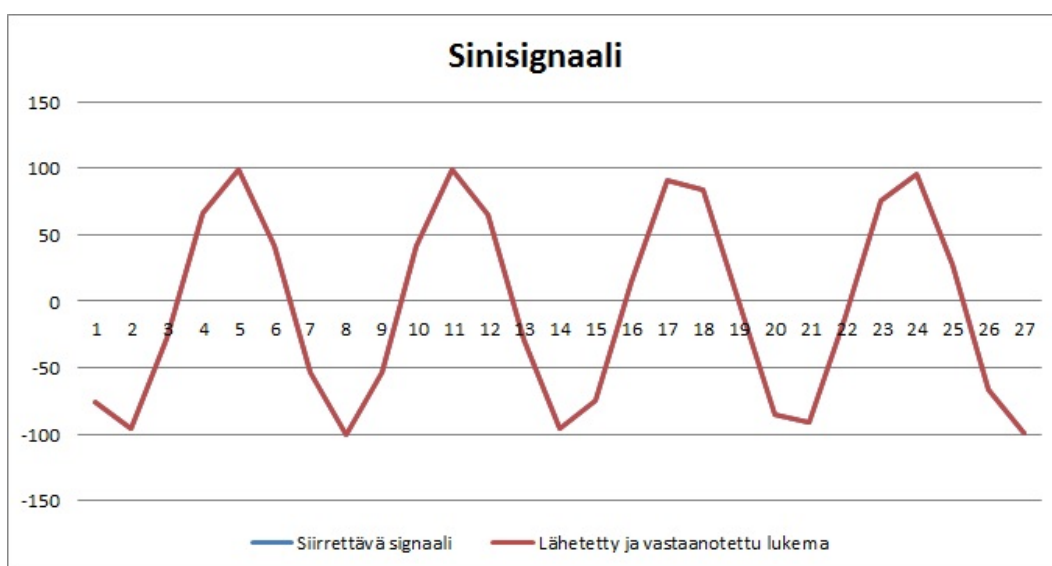
Taulukko 2. *Kantiaallon siirron ennustus.*

Lukema	Siirretty	Ennustettu
31	16	15

Kantiaallon siirtäminen onnistuu ennustamalla hyvin ja luotettavasti, kuten mallinnuksen perusteella osataan päätellä. Taulukon (Taulukko 2) Lukema sarakkeen lukumäärä on sama kuin verkon yli lähetettyjen sanomien määrä.

7.2.2 Sinitesti

Siniaaltoa vastaavaa signaalia vastaavat arvot tallennettiin Arduinon muistikortille rivikohtaisina lukemina ja ladattiin alkulatauksen yhteydessä mikrokontrollerin muistissa olevaan taulukkoon. Ohjelma simuloi mittausdataa käymällä läpi syklisesti taulukon arvoja.



Kuvio 31. Siniaallon siirtäminen ennustamalla

Luettu tieto ja vastaanotettu tieto tasmäsivät täydellisesti toisiinsa (Kuvio 31), joten kuvaajan viivat asettuivat nätisti päällekkäin eikä yksikään mittaustulos jäänyt siirtämättä kuten tuloksista (Taulukko 3) nähdään.

Taulukko 3. *Siniaallon siirron ennustus.*

Lukema	Siirretty	Ennustettu
27	27	0

Siirtoon suunniteltu protokolla toimi oletuksen mukaisesti ja ennustamista ei tapahtunut.

7.2.3 Linearisesti muuttuva signaali

Lineaarista muutosta edustavaa signaalia vastaavat arvot tallennettiin Arduinon muistikortille rivikohtaisina lukemina ja ladattiin alkulatauksen yhteydessä mikrokontrollerin muistissa olevaan taulukkoon. Ohjelma simuloi mittaustietoa käymällä syklisesti läpi taulukon arvoja.



Kuvio 32. Lineaarisen signaalin siirtäminen ennustamalla

Kuten kuvaajasta (Kuvio 32) nähdään, lineaarisessa muutoksessa algoritmi tunnistaa signaalin ja protokolla lähettää vain tarvittavat tiedot vastaanottajalle. Vastaanotetuista tiedoista saadaan signaalille malli jolla se voidaan palauttaa alkuperäistä vastaavaksi. Tulokset (Taulukko 4) osoittavat että ennustus sopii hyvin tämänkaltaiselle signaalille.

Taulukko 4. Lineaarisesti muuttavan signaalin siirron ennustus.

Lukema	Siirretty	Ennustettu
36	17	19

Tämänkaltaisella signaalilla ennustamisen avulla voidaan jo jättää lähettämättä puolet tiedoista, ja ideaaliolosuhteissa alkuperäinen signaali saadaan mallinnettua täydellisesti.

7.2.4 Säämittaustieto

Säämittaustietona käytetään ilmatieteenlaitoksen auringonsäteilyn globaalien säteilyn mitaustietoa. Ilmatieteenlaitos (2016) määrittelee globaalien säteilyn seuraavasti :

"Globaalilla säteilyllä tarkoitetaan auringosta saapuvaa laajakaistaista lyhytaaltosäteilyä (W/m²), joka tulee anturille koko taivaankannelta."

Testiaineistovalinta sattui tähän helpon saatavuutensa sekä minuutin syklissä tapahtuvan tallennuksen vuoksi, eli se on hyvä verrokkiksi normaalille mitaustapahtumalle. Lisäksi kyseessä on jatkuvasti vaihtuva arvo muun muassa pilvistä johtuen. Testissä halutaan käyttää todellista

"elävää" lukemaa ja todeta miten algoritmi toimii sellaisen kanssa.

Testissä käytetyn aineiston saa itselleen rekisteröitymällä Ilmatieteenlaitoksen avoimen datan palveluun ja siirtämällä sieltä halutun tiedon itselleen. Tiedon käyttäminen vaatii rekisteröitymisen palveluun ja on yksityisille ilmaista. Tutkielmassa käytetyn auringonsäteilyn mittaustiedon Jyväskylästä saa itselleen osoitteesta:

http://data.fmi.fi/fmi-apikey/SYÖTÄ_OMA_API_AVAIMESI_TÄHÄN/wfs?request=getFeature&storedquery_id=fmi::observations::radiation::timevaluepair¶meters=glob_1min&fmid=101339&starttime=2016-11-10T00:00:00Z&endtime=2016-11-11T00:00:00Z

Kyselyssä voi määritellä halutun suureen ja parametreilla saa takemmin määrättyä alkamisajankohdan (starttime), päättymisajankohda (endtime) sekä sijainnin (fmid). Ajankohta, joka siirretään mikrokontrollerista palvelimelle, alkaa 10.11.2015 kello 0:00 ja päättyy siitä 600 minuutin kuluttua. Haetussa datassa alue oli suurempi, mutta itse testiin tuli taulukon koon takia pienempi mittausalue.

Taulukko 5. Auringonsäteilysignaalin siirron ennustus.

Lukema	Siirretty	Ennustettu
573	25	2

Testien tuloksesta (Taulukko 5) käy ilmi ettei auringon säteily anna hyviä tuloksia tällä algoritmilla. Edestakaisin muuttuva signaali aiheuttaa lisäksi kulmakertoimen kautta virheellisiä ennustesarvioita, koska algoritmiä ei ole opetettu tunnistamaan kahden arvon välillä vaihtelevaa arvoa. Mittaustoleranssin salliminen lisäisi mahdollisia ennusteita, mutta koska testissä mittaustoleranssi on nolla, ei epätarkkaa ennustamista sallita.

Algoritmi säästää noin 4% siirtokapasiteettia auringonsäteilyn mittauksessa. Tulos on konkavisuudessaan rohkaiseva, koska ennustamiseen käytettävä algoritmi on yksinkertainen. Tehokkaammalla mallinnuksella sekä ennustusmenetelmällä tulos todennäköisesti parantuu.

7.3 Testien ongelmat

Testit onnistuivat hyvin pienistä protokollan ohjelmointiin liittyvistä ongelmista huolimatta. Testijärjestely osoittautui toimivaksi kun oletuksena on, ettei ulkopuolisia häiriötekijöitä ole ja mittaustulos on virheetöntä. Oikealla mittaustiedolla edestakaisesti muuttuva signaali aiheutti virheellisen ennustuksen, mutta virhe on tunnistettavissa helposti saapuneesta tiedosta ja voidaan opettaa järjestelmälle. Virhe johtuu PNS-menetelmän laskennasta jossa kaksi arvoa vaihtelee keskenään. Tässä tilanteessa suoran kulmekerroin pysyy samana koska vaihtelu tapahtuu tasaisesti suoran kummallakin puolella.

Kokonaisuutena testaaminen onnistui hyvin ja ilman suurempia ongelmia. Onnistuminen johtui hyvästä esisuunnittelusta, algoritmien perusteellisesta pöytätestaamisesta ja selkeästä ratkaisumallista. Näistä johtuen ongelmakohdat olivat helposti paikannettavissa ja korjattavissa.

8 Analysointi

Tässä tutkielmassa on tarkoitus selvittää voidaanko tietojensiirtoon käytettäviä resursseja säästää käyttämällä älykkäämpiä tiedon tunnistavia siirtoprotokollia. Tutkielmassa rakennettiin ennustava protokolla ja protokollaa simuloitiin rakennetuilla testisignaaleilla ja oikealla mittaustiedolla.

8.1 Kanttiaalto

Kanttiaallon siirrossa menetelmä on varsin tehokas ja jopa varsin hyvä. Menetelmä ei tee virheitä ja tilamuutokset saadaan siirrettyä lähes reaaliaikaisesti kuormittamatta silti tiedon-siirtoverkkoa. Tämä johtuu siitä että signaali on lineaarinen ja testissä käytetty menetelmä tunnistaa sen helposti.

Menetelmä soveltuu jo suoraan päällä/pois-tyyppisen tilatiedon siirtämiseen. Sen kokeileminen oikeassa ympäristössä akkukäyttöisillä laitteita olisikin testaamisen arvoista.

8.2 Sinitesti

Sinisignaalin siirto toimii oletusten mukaisesti. Sinisignaalilla ennustamisen ei pitänyt olla mahdollista, koska siinä ei ole lineaarista osuutta. Siirtotesti osoittaa sen testitilanteessa. Sinikomponenttia sisältäviin mittauksiin testattu menetelmä ei sovellu ennenkuin signaalimuoto on opetettu järjestelmälle.

8.3 Linearisesti muuttuva signaali

Linearisesti muuttuvalla signaalilla menetelmä toimii mallinnuksen mukaan hyvin ja luotettavasti. Lähettävän laitteen algoritmilla tunnistetaan lineaarinen muutos ja vastaanotetusta tiedosta alkuperäinen signaali saadaan palautettua täydellisesti. Menetelmä soveltuu rajoitettusti tämän tyyppiselle signaalille, kunhan sille sallitaan mittausvirhettä tehostamaan menetelmää entisestään.

8.4 Säädä

Säädässa menetelmä toimii oletusten mukaan. Huolimatta muutamista virheistä, signaalin tunnistaminen toimii ja virhetulkintojen syyt on tunnistettavissa. Tämän kaltainen signaali vaatisi tehokkaampaa ennustamismenetelmää tai suurempaa sallittua virhettä siirretylle tiedolle. Resurssien säästämisen suhteen testattu menetelmä ei suoriudu kovin tehokkaasti mutta tulokset ovat lupaavia jatkokehitystä ajatellen.

9 Yhteenveto

Tutkielma osoitti että jo olemassa olevilla tekniikoilla voidaan mallintaa signaalin luonnetta luotettavasti sen käyttäytymisen ennustamiseksi. Tällä tavalla on mahdollista säästää verkon siirtokapasiteettia ja laitteiden resursseja niin tallennustilan kuin laskennan suhteen. Käytännössä tämä tarkoittaa muun muassa pidempiä valmiusaikoja akkulaitteille, nopeampaa tiedonsiirtoa ja verkon kuormituksen pienenemistä. Kaikkein tärkeimmäksi näistä voivat tulevaisuudessa muodostua yksittäisen mittalaitteen akun kestoa pidentävät vaikutukset ja mittauksen tosiaikaisempi tiedonsiirto.

Jatkotutkimuksen tarpeena on tehokkaampien ennustamismenetelmien etsiminen ja koneoppimisen optimointi usealle eri menetelmälle. Menetelmiä parantamalla päästään tilanteeseen, jossa tunnetut rakenteet tunnistetaan ja voidaan mallintaa vastaanottopäässä luotettavasti. Näillä keinoin saadaan mm. sinikomponentti ja pulssitukset toistettua ilman varsinaista tiedonsiirtoa, tai mahdollisesti toistettua edellisen vuorokauden mittaustulokset kokonaan ennustamalla. Muita kehittämiskohteita ovat mittausvirheen käsittelyyn ja kommunikoinnin virheiden havaitsemiseen kohdistuvat kehitystarpeet.

Vaikka ennustavan menetelmän tulos ei oikealla mittaustiedolla ollut suuri, on tuloksen suunta oikea ja vaikutuksen suuruudenkin arviointi suhteellista. Mikäli nykyisten laitteiden 10 vuoden akkunkestoa voidaan tämänkaltaisella menetelmällä kasvattaa 4%:ia, kokonaisvaikeus on melkein puolivuotta lisää valmiusaikaa.

Lähteet

- BIPM. 2016. “About the BIPM”. Viitattu 25. marraskuuta 2016. <http://www.bipm.org/en/about-us/>.
- Blum, Avrim L, ja Pat Langley. 1997. “Selection of relevant features and examples in machine learning”. *Artificial intelligence* 97 (1): 245–271.
- Brockwell, Peter J, ja Richard A Davis. 2002. *Introduction to time series and forecasting*. Springer Science & Business Media.
- Cheung, H., ja S. Mulvey. 2014. “Great miscalculations: The French railway error and 10 others”. Viitattu 25. marraskuuta 2016. <http://www.bbc.com/news/magazine-27509559>.
- CIA. 2016. “The World Factbook”. Viitattu 25. marraskuuta 2016. <https://www.cia.gov/library/publications/the-world-factbook/appendix/appendix-g.html>.
- CompuServe Incorporated. 1987. *Graphics Interchange Format (GIF) Specification*. 15. kesäkuuta. <http://www.w3.org/Graphics/GIF/spec-gif87.txt>.
- Decotignie, J-D. 2005. “Ethernet-based real-time and industrial communications”. *Proceedings of the IEEE* 93 (6): 1102–1117.
- Dunn, Christopher. 1986. “Smile! You ’re on RLE!” *The Transactor* 7 (06): 16–18. http://csbruce.com/cbm/transactor/pdfs/trans_v7_i06.pdf.
- Felser, Max. 2005. “Real-time ethernet–industry prospective”. *Proceedings of the IEEE* 93 (6): 1118–1129.
- Gartner, Inc. 2014. “Gartner Newsroom”. Viitattu 25. marraskuuta 2016. <http://www.gartner.com/newsroom/id/2905717>.
- . 2016. “Gartner Newsroom”. Viitattu 25. marraskuuta 2016. <http://www.gartner.com/newsroom/id/3412017>.

Hartley, Ralph VL. 1928. "Transmission of information1". *Bell System technical journal* 7 (3): 535–563.

Heidegger, Martin. 1977. "The age of the world picture". *The question concerning technology and other essays* 1:115–154.

Hsu, Hwei P, ja Rakesh Ranjan. 2014. *Signals and Systems*. McGraw-Hill.

Ilmatieteenlaitos. 2016. "Havaintosuureet". Viitattu 14. marraskuuta 2016. <http://ilmatieteenlaitos.fi/havaintosuureet>.

Investopedia, LLC. 2017a. "Exponential Moving Average - EMA". Viitattu 12. tammikuuta 2017. <http://www.investopedia.com/terms/e/ema.asp>.

———. 2017b. "Simple Moving Average - SMA". Viitattu 12. tammikuuta 2017. <http://www.investopedia.com/terms/s/sma.asp>.

———. 2017c. "What's the difference between moving average and weighted moving average?" Viitattu 12. tammikuuta 2017. <http://www.investopedia.com/ask/answers/071414/whats-difference-between-moving-average-and-weighted-moving-average.asp>.

Jerri, Abdul J. 1977. "The Shannon sampling theorem—Its various extensions and applications: A tutorial review". *Proceedings of the IEEE* 65 (11): 1565–1596.

Järvinen, Petteri. 2003. *IT-tietosanakirja*. Docendo.

Keogh, Eamonn, Selina Chu, David Hart ja Michael Pazzani. 2001. "An online algorithm for segmenting time series". Teoksessa *Data Mining, 2001. ICDM 2001, Proceedings IEEE International Conference on*, 289–296. IEEE.

Kopetz, Hermann. 1991. "Event-triggered versus time-triggered real-time systems". Teoksessa *Operating Systems of the 90s and Beyond*, 86–101. Springer.

Kurose, James F., ja Keith W Ross. 2013. *Computer Networking: A Top-Down Approach*. Financial Times/Prentice Hall.

Kvantitatiivisten menetelmien tietovaranto. 2016. “Regressioanalyysi”. Viitattu 20. joulukuuta 2016. <http://www.fsd.uta.fi/menetelmaopetus/regressio/analyysi.html>.

Laitinen, Esko, Mikko Mäkelä, Lauri Soininen ja Seppo Tuomola. 1990. *Tammertekniikan kaavasto*. Gummerrus kirjapaino OY.

Le Gall, Didier. 1991. “MPEG: A video compression standard for multimedia applications”. *Communications of the ACM* 34 (4): 46–58.

Lin, Jessica, Eamonn Keogh, Stefano Lonardi ja Bill Chiu. 2003. “A symbolic representation of time series, with implications for streaming algorithms”. Teoksessa *Proceedings of the 8th ACM SIGMOD workshop on Research issues in data mining and knowledge discovery*, 2–11. ACM.

Lin, Jessica, Sheri Williamson, Kirk Borne ja David DeBarr. 2012. “Pattern recognition in time series”. *Advances in Machine Learning and Data Mining for Astronomy* 1:617–645.

LNE. 2016. “History of measurement”. Viitattu 25. marraskuuta 2016. <http://www.french-metrology.com/en/history/history-measurement.asp>.

Mellin, Ilkka. 2007a. *Tilastolliset menetelmät Osa 4: Lineaarinen regressioanalyysi, Erityiskysymyksiä yleisen lineaarisen mallin soveltamisessa*. Viitattu 25. marraskuuta 2016. http://salserver.org.aalto.fi/vanhat_sivut/Opinnot/Mat-2.3128/IMLuennot07/TILRE100.pdf.

———. 2007b. *Tilastolliset menetelmät Osa 4: Lineaarinen regressioanalyysi, Johdatus regressioanalyysiin*. Viitattu 25. marraskuuta 2016. <https://math.tkk.fi/opetus/sovtoda/luennot/TILJR100.pdf>.

Michalski, Ryszard S, Jaime G Carbonell ja Tom M Mitchell. 2013. *Machine learning: An artificial intelligence approach*. Springer Science & Business Media.

MIT. 2016. *Gibbs' Phenomenon*. Viitattu 25. joulukuuta 2016. https://ocw.mit.edu/courses/mathematics/18-03sc-differential-equations-fall-2011/unit-iii-fourier-series-and-laplace-transform/operations-on-fourier-series/MIT18_03SCF11_s22_7text.pdf.

Mitchell, M., Tom. 1997. *Machine learning*. McCraw-Hill.

Modbus Organization. 2006. *MODBUS MESSAGING ON TCP/IP IMPLEMENTATION: IMPLEMENTATION GUIDE*. 24. lokakuuta. Viitattu 7. tammikuuta 2017. http://www.modbus.org/docs/Modbus_Messaging_Implementation_Guide_V1_0b.pdf.

———. 2017. “FAQ”. Viitattu 7. tammikuuta 2017. <http://www.modbus.org/faq.php>.

MPEG. 2016. “Who we are”. Viitattu 14. marraskuuta 2016. <http://mpeg.chiariglione.org/who-we-are>.

———. 2017. “MPEG Basics.” Viitattu 8. tammikuuta 2017. <http://mpeg.chiariglione.org/mpeg-basics>.

Neumann, Peter. 2007. “Communication in industrial automation—What is going on?” *Control Engineering Practice* 15 (11): 1332–1347.

Nyquist, Harry. 1928. “Certain topics in telegraph transmission theory”.

O’Leary, Amy. 2013. “An Honor for the Creator of the GIF”. *The New York Times, Bits*. <http://nyti.ms/1qgmt8f>.

Orfanidis, Sophocles J. 1995. *Introduction to signal processing*. Prentice-Hall, Inc.

Ortega, Antonio, ja Kannan Ramchandran. 1998. “Rate-distortion methods for image and video compression”. *IEEE Signal processing magazine* 15 (6): 23–50.

Patton, Michael Quinn. 1990. *Qualitative evaluation and research methods*. SAGE Publications, inc.

Picone, Joseph W. 1993. “Signal modeling techniques in speech recognition”. *Proceedings of the IEEE* 81 (9): 1215–1247.

PROFIBUS and PROFINET International. 2016. “System Description Technology and Application”. Viitattu 25. marraskuuta 2016. <http://www.profibus.com/nc/download/technical-descriptions-books/downloads/profinet-technology-and-application-system-description/download/20728/>.

- Pynnönen, Seppo. 2005. “Tilastollinen ennustaminen”. Viitattu 7. tammikuuta 2017. <http://lipas.uwasa.fi/~sjp/Teaching/Econometrics/Lectures/Examples/ennustaminen.pdf>.
- Quinlan, J. Ross. 1986. “Induction of decision trees”. *Machine learning* 1 (1): 81–106.
- Sadler, Christopher M, ja Margaret Martonosi. 2006. “Data compression algorithms for energy-constrained devices in delay tolerant networks”. Teoksessa *Proceedings of the 4th international conference on Embedded networked sensor systems*, 265–278. ACM.
- Shannon, Claude E. 1959. “Coding theorems for a discrete source with a fidelity criterion”. *IRE Nat. Conv. Rec* 4 (142-163): 1.
- . 2001. “A mathematical theory of communication”. *ACM SIGMOBILE Mobile Computing and Communications Review* 5 (1): 3–55.
- Siemens. 2008. *Tiedonsiirto teollisuudessa*. Viitattu 8. tammikuuta 2017. http://www.eis.fi/tapahtumat/2008/Siemens2008/Teollisuuden_tiedonsiirto_yleisesti_112008.pdf.
- Taanila, Aki. 2011. “Aikasarjaennustaminen”. Viitattu 22. marraskuuta 2017. <http://myy.haaga-helia.fi/~taaak/m/ennus.pdf>.
- Welch, Terry A. 1984. “A technique for high-performance data compression”. *Computer* 17 (6): 8–19.
- Wikipedia. 2016. “2010 Flash Crash”. Viitattu 22. marraskuuta 2016. https://en.wikipedia.org/wiki/2010_Flash_Crash.
- VTT. 2016. “Mittayksiköt”. Viitattu 22. marraskuuta 2016. <http://www.mikes.fi/mittayksik%C3%B6t>.