

SISÄLLYSLUETTELO

JOHDANTO	3
SUMEAT JÄRJESTELMÄT JA LOGIIKKA	4
SUMEA LOGIIKKA MIKROVERKKOJEN HALLINNASSA	5
ESIMERKKI MAHDOLLISESTA SUMEAN HALLINTA-JÄRJESTELMÄN TOTEUTUSTAVASTA	7
TOTEUTUS	8
Simulaation alkuarvot	9
Simulaatiosta ja sen kellosta	10
Testisignaali­lähteet	11
Akkusimulaattori	11
Virtalaskin	12
Virtojen ja akun varauksen normalisointi	13
Sumea logiikkapiiri	15
Hallintasi­gnaalien generointi	18
SIMULOINNIN TULOKSET	20
YHTEENVETO	27
LÄHDELUETTELO	28

JOHDANTO

Micro gridit eli mikroverkot tarkoittavat rajatun mittakaavan paikallisia sähköverkkoja, joita hallitaan paikallisesti ja jotka voivat toimia itsenäisesti pääverkoista erillään ongelmatilanteissa.

Koska mikrogridien on toimittava itsenäisesti isoista sähköverkoista, niiden täytyy hallita omaa tuotantoaan ja kulutustaan itsenäisesti, ja siten niiden hallintaan pätevät samat asiat kuin suurempien sähköverkkojenkin. (Berkeley Lab 2017)

SUMEAT JÄRJESTELMÄT JA LOGIIKKA

Sumeat järjestelmät ja logiikka mallintavat epävarmuutta tai -täsmällisyyttä logiikan ja joukko-opin avulla. Nämä tekniikat mahdollistavat luonnolliseen kieleen ja epätäsmällisiin tietoihin perustuvan päättelyn ohjelmallisesti. (Alander 2017)

Yleinen tapa käyttää sumeaa logiikkaa on kuvata jonkin asian johonkin joukkoon kuulmista liukuvana arvona tunnetulla välillä, yleensä $[0,1]$, jolloin arvo 1 tarkoittaa kuuluu täysin joukkoon ja 0 että asia ei kuulu joukkoon lainkaan. (Alander 2017)

Tällöin näihin lukuihin voidaan soveltaa seuraavia sumeita loogisia operaatioita.

- Sumea A Ja B: Pienempi luku A:sta ja B:stä.
- Sumea A Tai B: Suurempi luku A:sta ja B:stä.
- Sumea A:n Negaatio: $1-A$.
- Samankaltaisuus A:n ja B:n välillä: $1-|A-B|$

Ja ja tai operaattorit voidaan määritellä muillakin tavoilla, niiden halutuista ominaisuuksista riippuen. (Alander 2017)

SUMEA LOGIIKKA MIKROVERKKOJEN HALLINNASSA

Mikroverkkojen hallinnassa sumeaa logiikka käytetään muun muassa akkujen varaustason, käynnissä olevien generaattorityyppien ja energian oston ja myynnin hallintaan. (Saranya & Co 2015)(Chen & Co 2013)(Leonori & Co 2016)

Energiaa kulutetaan erilaisia määriä ajasta riippuen ja eri generaattorityypit tuottavat energiaa eritavoilla. Tämän vuoksi eri generaattoreita halutaan priorisoida eri tavoin.

Uusiutuvia energianlähteitä käyttäviä generaattoreita halutaan käyttää mahdollisimman paljon energiankäytön kattamiseksi, mutta niiden tuotto vaihtelee suuresti ympäristöstä riippuen. (Saranya & Co 2015)(Chen & Co 2013)

Erilaisia polttoaineita käyttävät generaattorit tuottavat tasaisesti sähköä, mutta ne tuottavat yleensä kasvihuonekaasuja ja polttoaine maksaa. Joten niitä on suositeltavaa käyttää mahdollisimman vähän ja vain energiavajeen kattamiseksi. (Saranya & Co 2015)(Chen & Co 2013)

Energian varastointi järjestelmiä, kuten akkuja ja hydrolyysi/vedynpoltto järjestelmiä voidaan käyttää energian ylituotannon väliaikaiseen varastointiin. Mutta kaikki energiansäilytysjärjestelmät muuttavat energiaa sähköstä johonkin toiseen muotoon, mikä aiheuttaa energiahäviöitä. Lisäksi akkujen elinikä riippuu niiden varaustasosta, joten niiden varaustasoa kannattaa yrittää pitää akun tunnetulla optimaalisella tasolla jos mahdollista. (Saranya & Co 2015)(Chen & Co 2013)(Leonori & Co 2016)

Jos mikroverkko on liitetty pääsähköverkkoon on lisäksi hallittava milloin pääsähköverkosta ostetaan sähköä ja milloin sitä myydään sinne. Parhaat ajat tälle tietenkin riippuvat pääverkon senhetkisistä osto- ja myyntihinnoista sekä mikroverkon energiavaraston ja sen hetkisen tuotannon omasta tilanteesta. Energiaa halutaan ostaa, kun mikroverkko ei kykene kattamaan omaa kulutustaan tai jos tarjolla on erityisen halpaa energiaa ja energiavarastot ovat optimaalisesta tasosta vajaita. Ja sitä halutaan

myydä, kun mikroverkko tuottaa enemmän kuin se kuluttaa ja energiavarastot ovat jo optimaalisella tasolla tai mahdolliset tulot ovat erityisen suuria. (Leonori & Co 2016)

Näiden kaikkien muuttujien ja kohteiden yhtäaikainen hallinnointi pelkällä binäärisellä logiikalla on vaikeaa, esimerkiksi päätös siitä kannattaako energian ylituotantoa myydä vai varastoida juuri tällä hetkellä riippuu siitä kuinka lähellä energiavarastot ovat optimaalista tasoa ja kuinka korkeaa hintaa pääverkko tarjoaa. Sumean logiikan menetelmät ovat suunniteltu juuri tällaisten epämääräisten päätösten tekemiseen.

ESIMERKKI MAHDOLLISETA SUMEAN HALLINTA-JÄRJESTELMÄN TOTEUTUSTAVASTA

Mikroverkon sumean hallintajärjestelmä voisi toteuttaa esimerkiksi siten että mikroverkon senhetkinen uusiutuva tuotanto kulutukseen verrattuna, kokonaistuotanto kulutukseen verrattuna, nykyisten energiavarastojen taso optimaaliseen verrattuna ja pääverkon tarjoamat hinnat muutetaan sumeiksi arvoiksi tosi pieni, pieni, noin nolla, iso tai tosi iso. (Saranya & Co 2015)(Chen & Co 2013)(Leonori & Co 2016)

Tämän jälkeen voidaan käyttää esimerkiksi seuraavanlaista sumeisiin loogisiin operaattoriin perustuvaa sääntökantaa joka perustuu osittain Saranyan & kumppaneiden (2015), Chenin & kumppaneiden (2013) ja (Leonori & Co 2016) toteutuksiin, ja osittain omaan mietiskelyyn.

- Jos kokonaistuotto/kulutus on (iso TAI tosi iso) JA varastot on (pieni TAI tosi pieni), niin varastoi energiaa.
- Jos kokonaistuotto/kulutus on (iso TAI tosi iso) JA ((varastot on pieni JA myyntihinta on tosi iso) TAI (varastot on (noin nolla TAI (iso TAI tosi iso)))), niin myy energiaa.
- Jos uusiutuva tuotto/kulutus on (pieni TAI tosi pieni) JA varastot on (noin nolla TAI (iso TAI tosi iso)), käytä varastoja.
- Jos uusiutuva tuotto/kulutus on (pieni TAI tosi pieni) JA varastot on tosi pieni, käytä polttoaine generaattoreita.
- Jos (varastot on (pieni TAI tosi pieni) JA ostohinta on tosi pieni) TAI (kokonaistuotto/kulutus on (pieni TAI tosi pieni) JA varastot on tosi pieni), osta energiaa.

TOTEUTUS

Edellisen kappaleen esimerkkiä lähtökohtana käyttäen toteutettiin Scilab ja Xcos ohjelmilla seuraavanlainen yksinkertaistettu simulaatiomalli mikrogridin sumeasta hallintajärjestelmästä. Kuviossa akkusimulaattori ja erinäiset testisyötteiden signaalilähteet on korostettu violetilla, virtalaskin vaaleansinisellä, normalisointilaskimet keltaisella, varsinainen sumea päätöksentekijä vihreällä ja signaalilähteiden hallintasyötteiden generointi punaisella.


```

MaxGenOutput=2; //Generaattorin maksimituotto
MaxRenOutput=1; //Uudistuvien maksimituotto
MaxUseDraw=8; //Maksimikäyttö
MaxAbsGrdFlow=2; //Pääverkon osto/myynti siirtonopeus
MaxAbsBatFlow=1; //Akun siirtonopeus

//Kokonaisvirran edellisiin arvoihin perustuvat minimi-
//ja maksimiarvot
MinTotalFlow=(-MaxUseDraw-MaxAbsGrdFlow-MaxAbsBatFlow);
MaxTotalFlow=(MaxGenOutput+MaxRenOutput+MaxAbsGrdFlow+M
axAbsBatFlow);

BatMaxCap=200; //Akun maksimikapasiteetti
BatMinCap=0; //Akun minimikapasiteetti
BatEfficiency=0.75; //Akun tehokkuus
BatOptLvl=100; //Akun optimi varaustaso

//Sumeat luokat
BigMinus=-0.66; //Tosi pieni
SmallMinus=-0.33; //Pieni
AboutZero=0; //Noin nolla
SmallPlus=0.33; //Iso
BigPlus=0.66; //Tosi iso

RenPhase=0; //Uudistuvan tuoton vaihe
RenFreq=0.1; //Uudistuvan tuoton taajuus

UsePhase=1; //Kulutuksen vaihe
UseFreq=0.05; //Kulutuksen taajuus

SimLen=1000; //Simulaation pituus
ModeTresh=0.33; //Moodin vaihtokynnys

```

Simulaatiosta ja sen kellosta

Simulaation kellon peroidina käytettiin Xcosin oletusarvoa 0.1, ja sen pituus määriteltiin SimLen vakion perusteella. Hinnanmuutosten ja hallintasiinaalien arvot mitattiin SimLen vakion sadasosan välein. Simulaatiomalleissa esiintyvä Z-signaali

vastaa nollaa, sen vakiolähde on erotettu kaaviossa kellopalikoiden joukkoon turhien signaalilinkkien välttämiseksi.

Testisignaali-lähteet

Simulaation testaukseen käytettiin seuraavanlaisia testisignaali-lähteitä. Uudistuvan tuoton ja kulutuksen simulointiin käytettiin siniaaltoja joiden taajuutta, vaihetta ja amplitudia säädettiin vastaavilla vaihe,taajuus ja maksimituotto/käyttö vakioilla.

Generaattorisignaalinä käytettiin nollaa tai sen maksimituottoa vastaavaa vakiota joiden välillä vaihdettiin logiikkapiirin palauttaman hallintasiignaalin perusteella. Jos hallintasiignaali oli positiivinen ja suurempi kuin moodin vaihdon kynnsarvo, generaattoria ajettiin, muuten se jätettiin pois päältä.

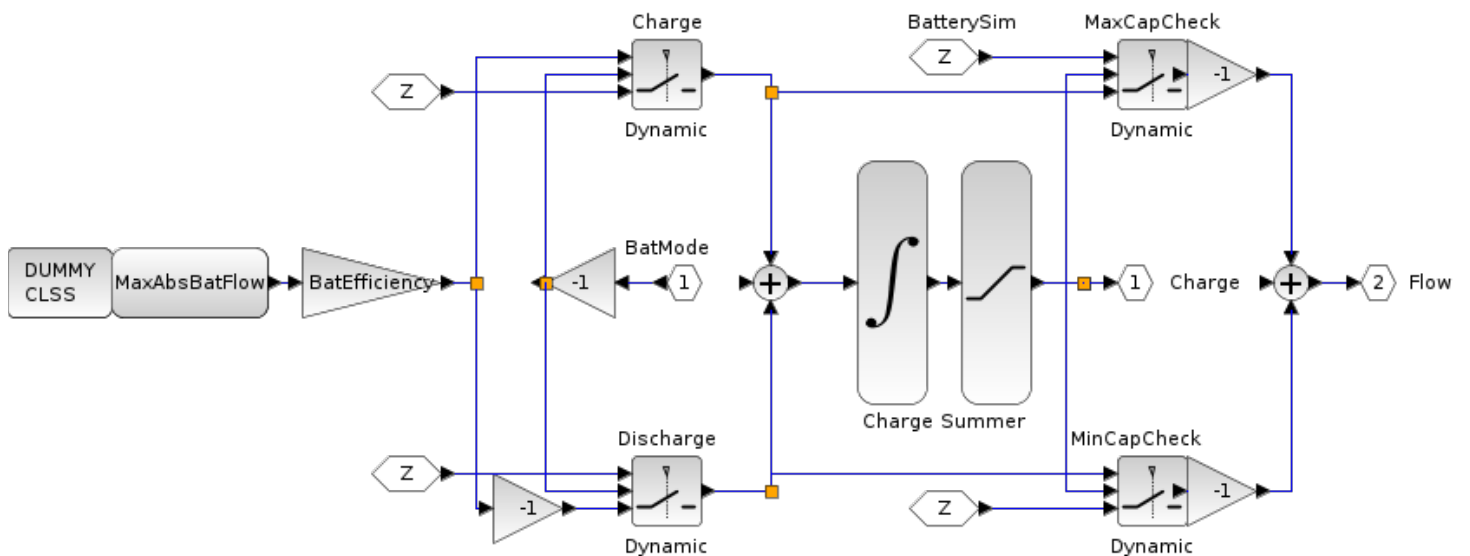
Verkkosignaalin toteutus vastasi generaattorisignaalin toteutusta, paitsi että sen antama arvo oli joko positiivinen vakio, negatiivinen vakio tai nolla, hallintasiignaalin merkistä ja kynnsarvon ylityksestä riippuen.

Osto ja myyntihintojen signaalinä käytettiin samaa signaalia, joka generoitiin satunnaisgeneraattorilla, sen arvo mitattiin mittauksen pituuden sadasosan välein. Satunnaisgeneraattori määriteltiin tuottamaan lukuja välillä $[-1,1]$, joten hintoja ei tässä simulaatiossa tarvinnut normalisoida erikseen.

Akkusimulaattori

Akun tuottama signaali perustui seuraavan kuvion mukaiseen yksinkertaistettuun akkusimulaattoriin, joka on suunniteltu itse yritys ja erehdys periaattella. Sen ainut tarkoitus oli tuottaa jotain suurin piirtein akun kaltaista käyttäytymistä, ei simuloida oikeaa akkua.

Akuusimulaattori toimii siten että hallintasiignaalin merkin mukaan akkua joko ladataan tai puretaan MaxAbsBatFlow ja BatEfficiency vakioiden tuloon määrittämällä nopeudella. Latausmoodissa simulaattori kasvattaa varauksen arvoa ja palauttaa negatiivista virtaa, kunnes se saavuttaa maksimikapasiteettinsa, purkumoodissa se toimii päinvastaisesti. Huomaa että hallintasiignaalin etumerkki vaihdetaan ennen sen käyttöä akun hallintaan. Tämä siksi että koko simulaation näkymässä negatiivinen merkki esittää akun latausta ja positiivinen purkua, jotta ne vastaisivat akkusimulaattorin tuottamaa virtaa. Kun taas akkusimulaattorin sisällä negatiivinen merkki esittää purkua ja positiivinen latausta, jotta ne vastaisivat akun varauksen laskua ja nousua.

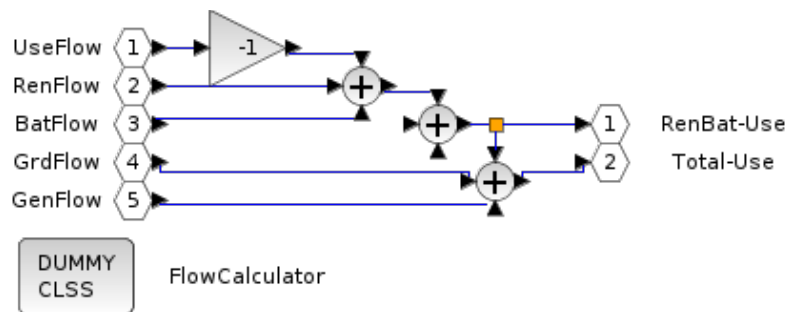


Kuvio 2. Akkuusimulaattorin malli.

Virtalaskin

Toisin kuin lähteissä ja neljännen kappaleen yksinkertaistetussa esimerkissä, seuraavassa kuviossa näkyvä toteutettu virtalaskin laskee tuoton ja kulutuksen välisen eron, sen sijaan että jakaisi tuoton kulutuksella. Tämä sen vuoksi, että akun ja verkkovirran vaihtelevista merkeistä johtuen, erotuksen laskeminen oli helpompaa kuin niiden ohjaaminen jakoviivan oikealle puolelle.

Lisäksi kyseisen erotuksen normalisointi oli helpompaa, koska kokonaisvirran mahdolliset minimi ja maksimiarvot oli helppo laskea tunnettujen lähdevirtojen minimi ja maksimiarvojen perusteella, kun taas osamäärän maksimiarvo saattoi kasvaa helposti laskettavissa olevien rajojen ulkopuolelle jos nimittäjän arvo päättyi $-1:n$ ja $1:n$ välille, muuttaen jakolaskun kertolaskuksi.



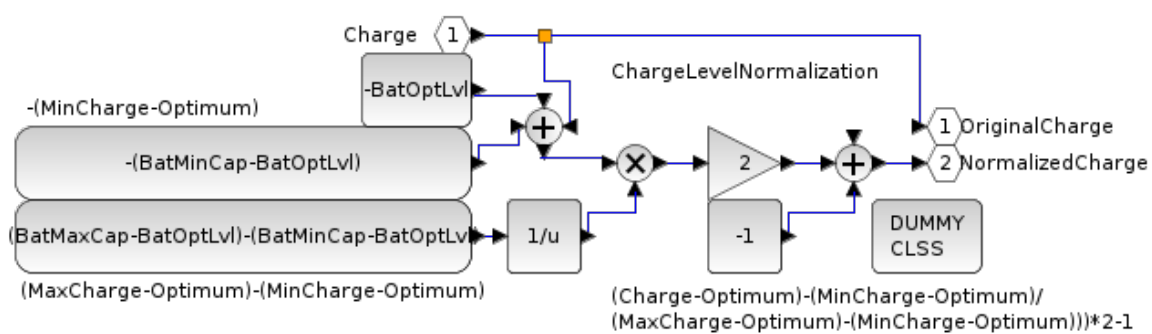
Kuvio 3. Virtalaskimen malli.

Virtojen ja akun varauksen normalisointi

Normalisointifunktiona käytettiin tavallista min max skaalausta välille $[-1,1]$. Eli seuraavanlaista kaavaa.

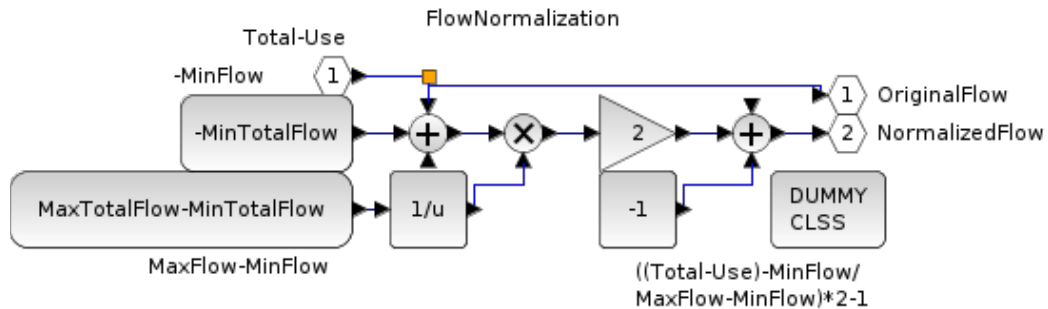
$$\frac{x - \text{Min}(x)}{\text{Max}(x) - \text{Min}(x)} * 2 - 1$$

Akun varauksen normalisoinnissa x :nä käytettiin varauksen ja optimaalisen varauksen erotusta optimaalisen varauksen sijoittamiseksi normalisoidun kuvaajan nollassolle, samoin myös $\text{Min}(x)$ ja $\text{Max}(x)$ arvoista vähennettiin optimaalisen varauksen arvo.



Kuvio 4. Akun varauksen normalisointifunktion malli.

Sekä kokonais- että uusiutuvan virran ja kulutuksen erotusten normalisointiin käytettiin samaa funktiota, jossa $x:n$ paikalla oli virtalaskimen antama tulos ja minimi ja maksimiarvoina käytettiin lähdevirtojen tunnettujen minimi ja maksimiarvojen summia.



Kokonaisvirtojen normalisointifunktion malli.

Sumea logiikkapiiri

Toteutettu sumea logiikkapiiri toimii siten että aluksi normalisoitujen virtojen ja varaustason arvot, sekä jo valmiina oikealla arvovälillä generoidut hintojen arvot, luokitellaan sumealla luokittelijalla seuraaviin luokkiin.

- Tosi iso
- Iso
- Noin nolla
- Pieni
- Tosi pieni

Tämän jälkeen käytetään seuraavanlaista, hieman kappaleen neljä esimerkistä muokattua sääntökantaa akun latauksen ja purun, generaattorien ajamisen sekä

pääverkon osto ja myynti signaalien generointiin. Max-funktiota on käytetty sumeana tai-funktiona ja Min-funktiota sumeana ja-funktiona.

$$LataaAkkua = \text{Min}(\text{Max}(PieniLataus, TosiPieniLatus), \\ \text{Max}(IsoKokonaisVirta, TosiIsoKokonaisVirta))$$

$$PuraAkkua = \text{Min}(\text{Max}(IsoLataus, TosiIsoLatus), \\ \text{Max}(PieniUusiutuvaVirta, TosiPieniUusiutuvaVirta))$$

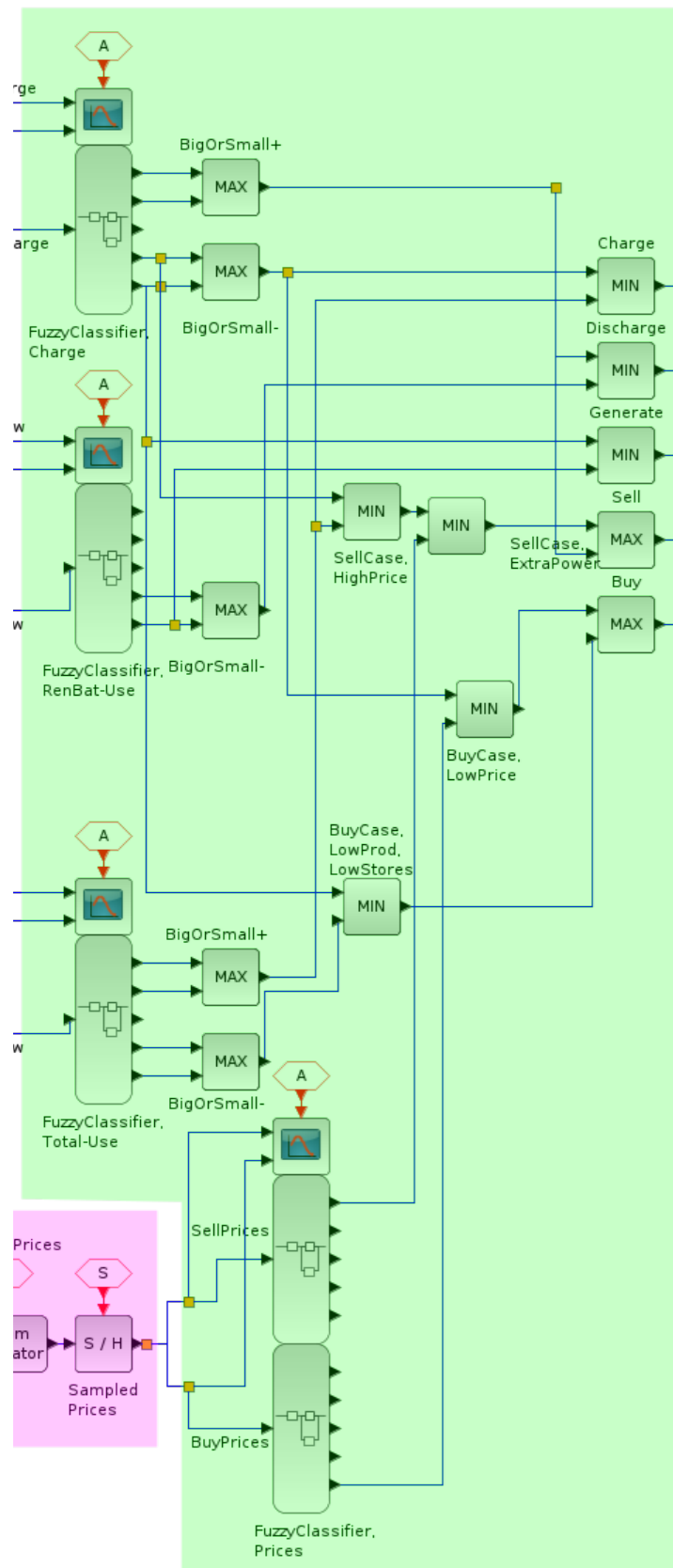
$$AjaGeneraattoria = \text{Min}(TosiPieniVaraus, TosiPieniUusiutuvaVirta)$$

$$Myy = \text{Max}(\text{Min}(\text{Min}(PieniVaraus, \text{Max}(IsoKokonaisVirta, TosiIsoKokonaisVirta))), TosiIsoMyyntiHinta), \\ \text{Max}(IsoVaraus, TosiIsoVaraus))$$

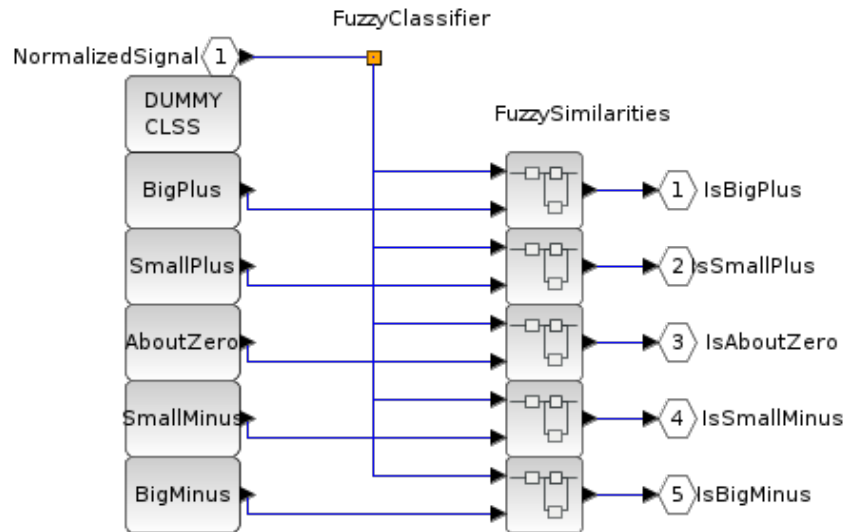
$$Osta = \text{Max}(\text{Min}(\text{Max}(PieniVaraus, TosiPieniVaraus), TosiPieniOstoHinta), \\ \text{Min}(TosiPieniVaraus, \text{Max}(PieniKokonaisVirta, TosiPieniKokonaisVirta)))$$

Huomaa että myyntifunktiossa korkean hinnan aikana myydään vain jos varaus on pieni ja kokonaisvirta on positiivinen. Tämä siksi että optimitason yli menevä varaus myydään aina ja akkua ei ole toivottavaa myydä kokonaan tyhjäksi.

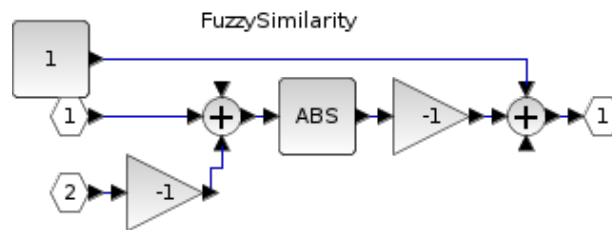
Simulaation toteutuksessa käytetty sumea luokittelija on yksinkertainen piiri, joka vertaa sisään tulevan normalisoidun arvon samankaltaisuutta kaikkien viiden sumean luokan vakioarvoihin ja palauttaa jokaisen vertailun tuloksen eri vasteeseen.



Kuvio 5. Sumean logiikkapiirin malli.



Kuvio 6. Sumean luokittelijan malli.



Kuvio 7. Sumean samankaltaisuuden malli.

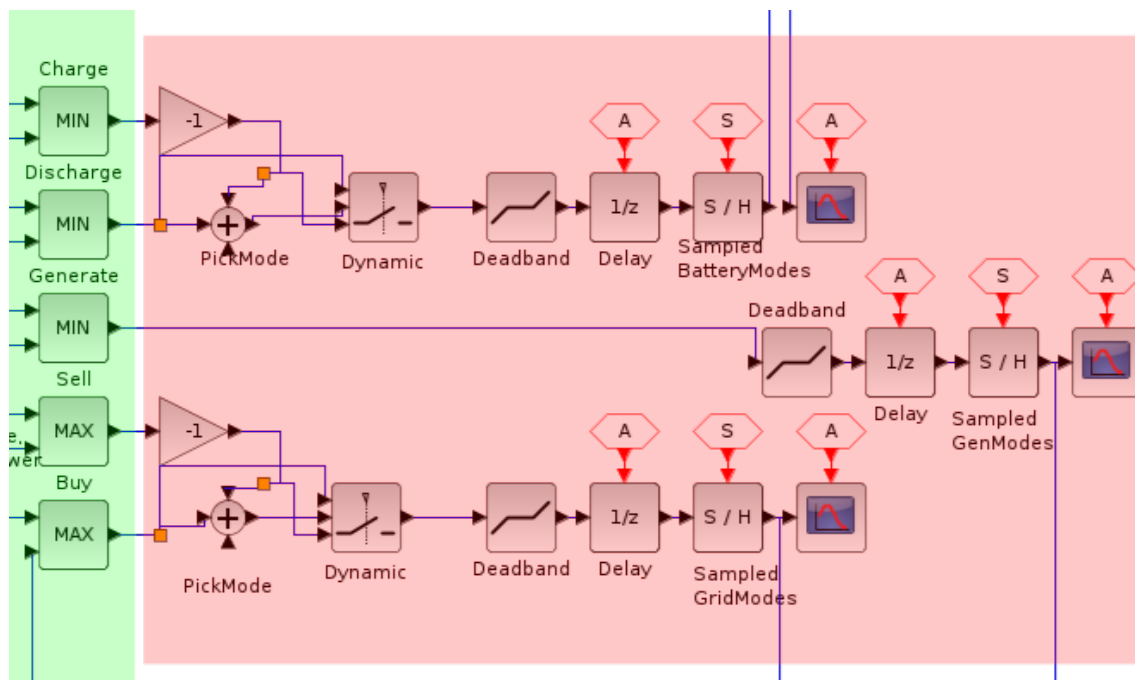
Hallintasiinaalien generointi

Akun lataus/purku moodin, pääverkon osto/myynti moodin ja generaattorin ajo moodin hallintasiinaalit generoidaan poistamalla deadbandilla sumean logiikkapiirin ulostuloista nollan ympärillä oleva alue, jonka koko on määritelty ModeTresh vakion avulla. Tämän jälkeen signaalia viivästetään yhdellä simulaatiokierroksella algebrasilmukoiden välttämiseksi ja sen arvo mitataan simulaation ajoajan sadasosan välein.

Akun varaus/purku ja pääverkon osto/myynti moodien kohdalla, logiikkapiirin antamista signaaleista valitaan suurempi ja jos kysessä on lataus tai myynti, niin sen etumerkki käännetään. Tämä tapahtuu ennen hallintasiinaalin generointia ja näkyy kuviossa 8 PickMode nimisenä pienpiirinä jonka tarkka kaava on seuraava.

$$PickMode(DischargeBuy, ChargeSell) = \begin{cases} DischargeBuy, & DischargeBuy - ChargeSell > 0 \\ -ChargeSell, & DischargeBuy - ChargeSell \leq 0 \end{cases}$$

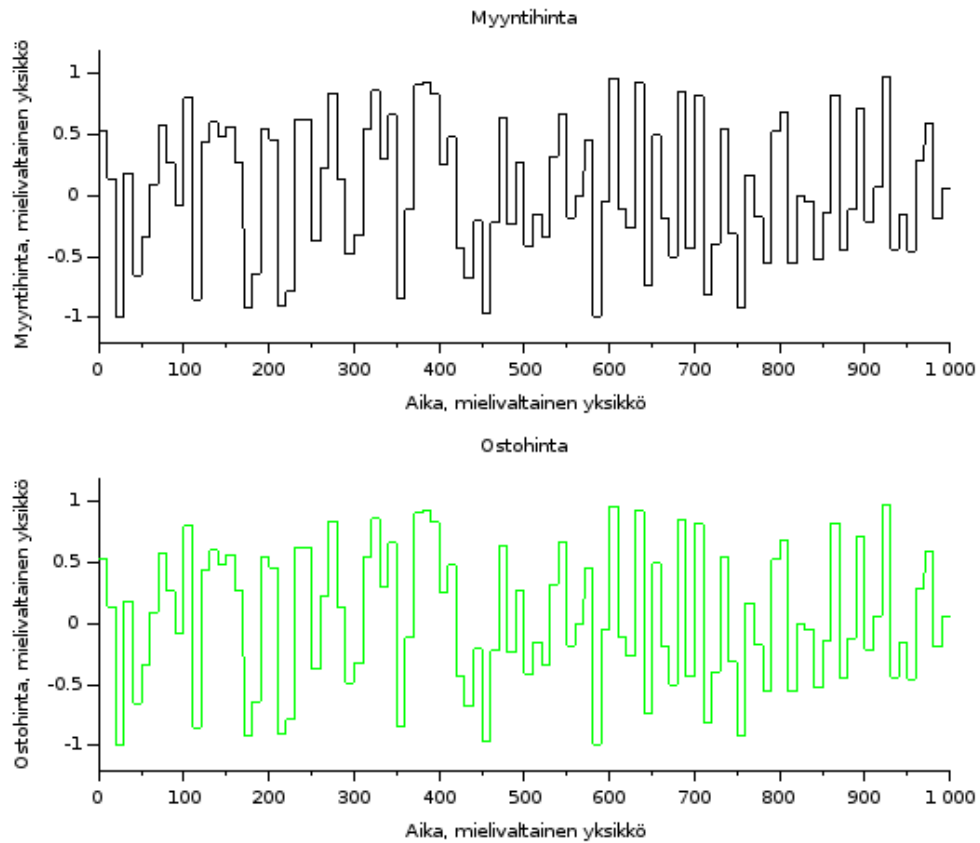
Tämä siksi että hallintasiinaalin etumerkki määrittelee akkusimulaattorin ja verkkosignaalieneraattorin moodin.



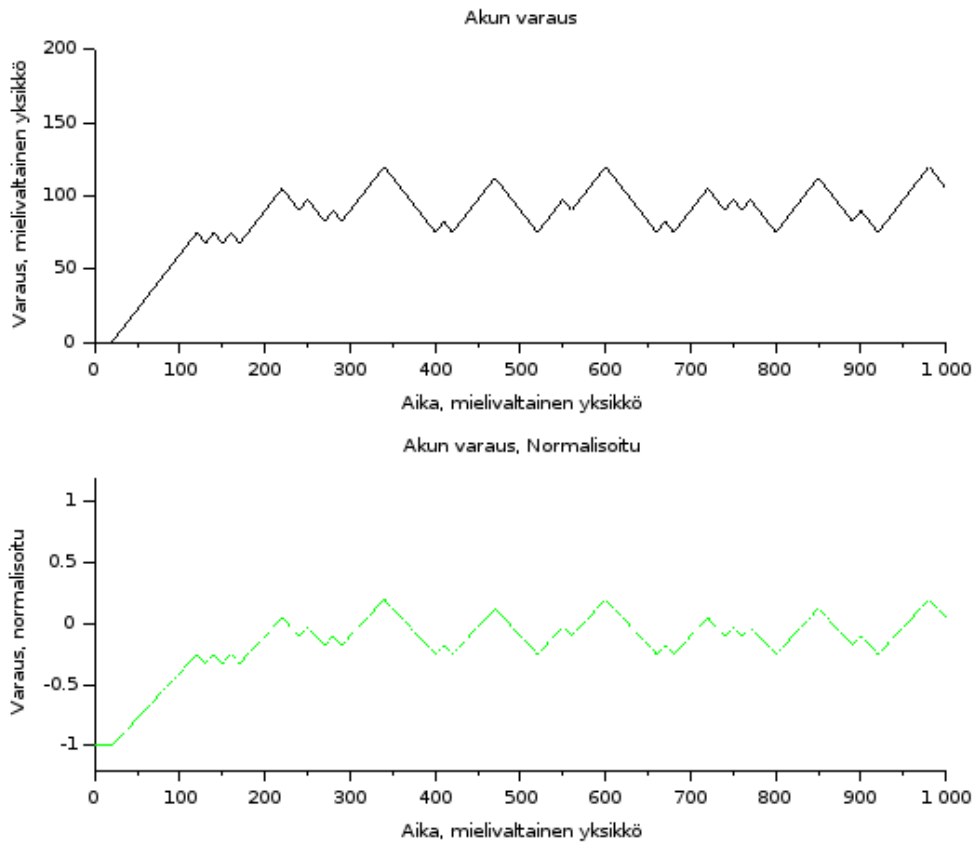
Kuvio 8. Hallintasiinaalien generoinnin malli.

SIMULOINNIN TULOKSET

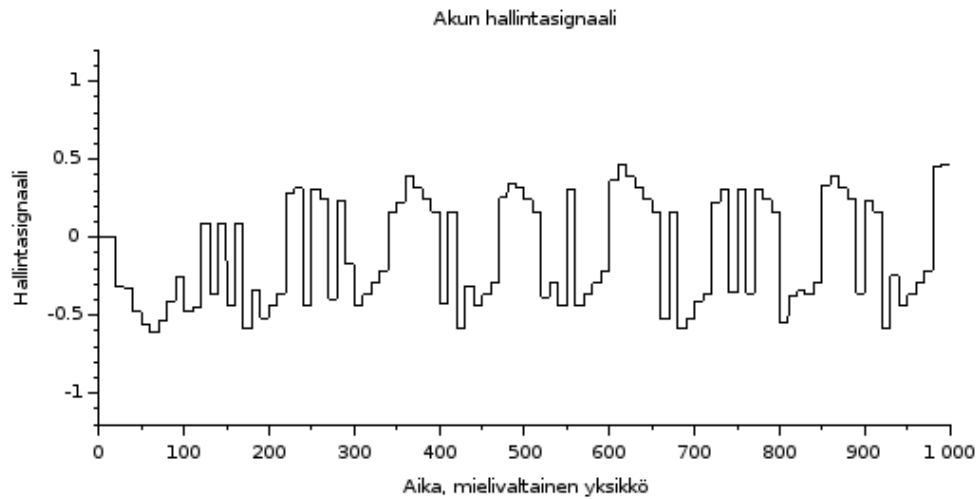
Tässä kappaleessa käsitellään simuloinnin tuloksia. Seuraavat kuvaajat esittävät hintojen, akun varauksen, hallintasiinaalien, ja tuotanton ja kulutuksen kuvaajia.



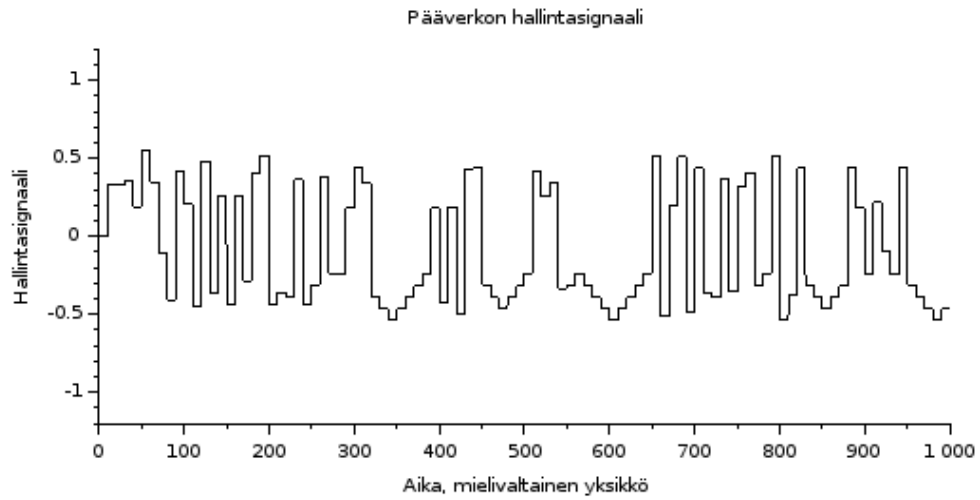
Kuvio 9. Hintojen kuvaajat, tässä simuloinnissa identtiset.



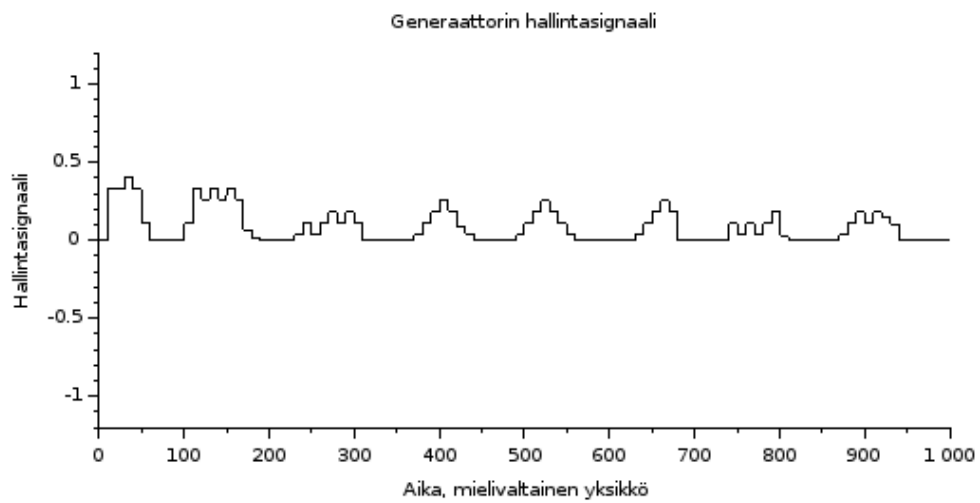
Kuvio 10. Akun varauksen kuvaajat.



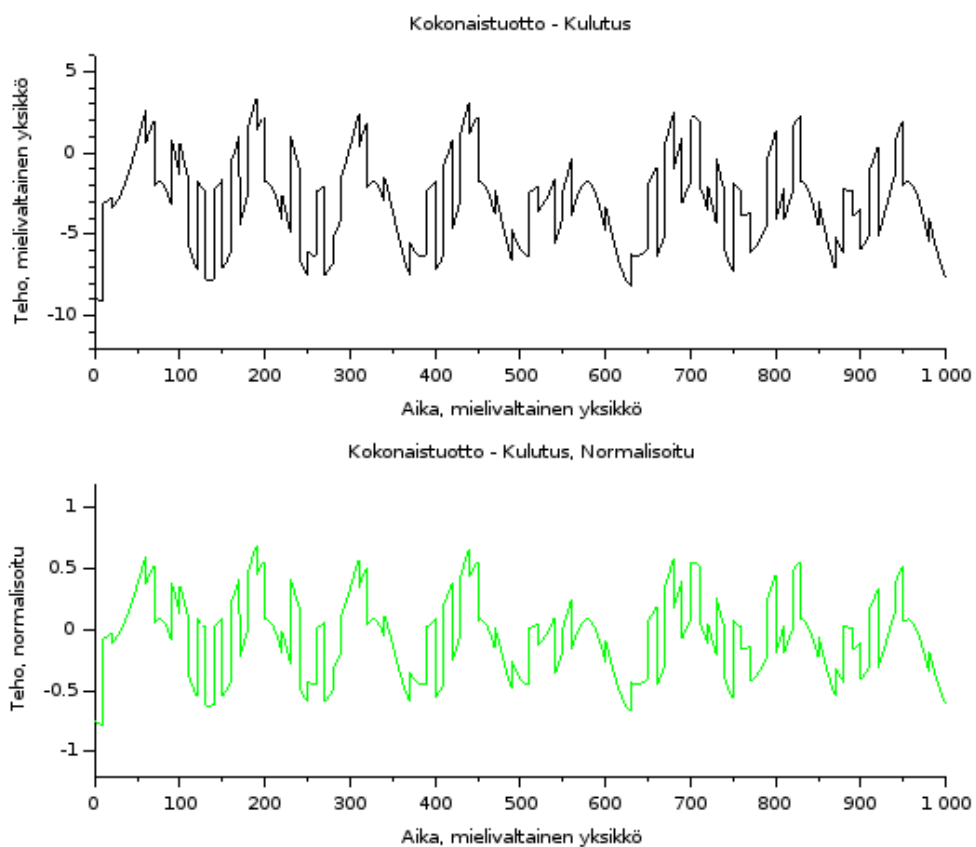
Kuvio 11. Akun hallintasiignaalin kuvaaja.



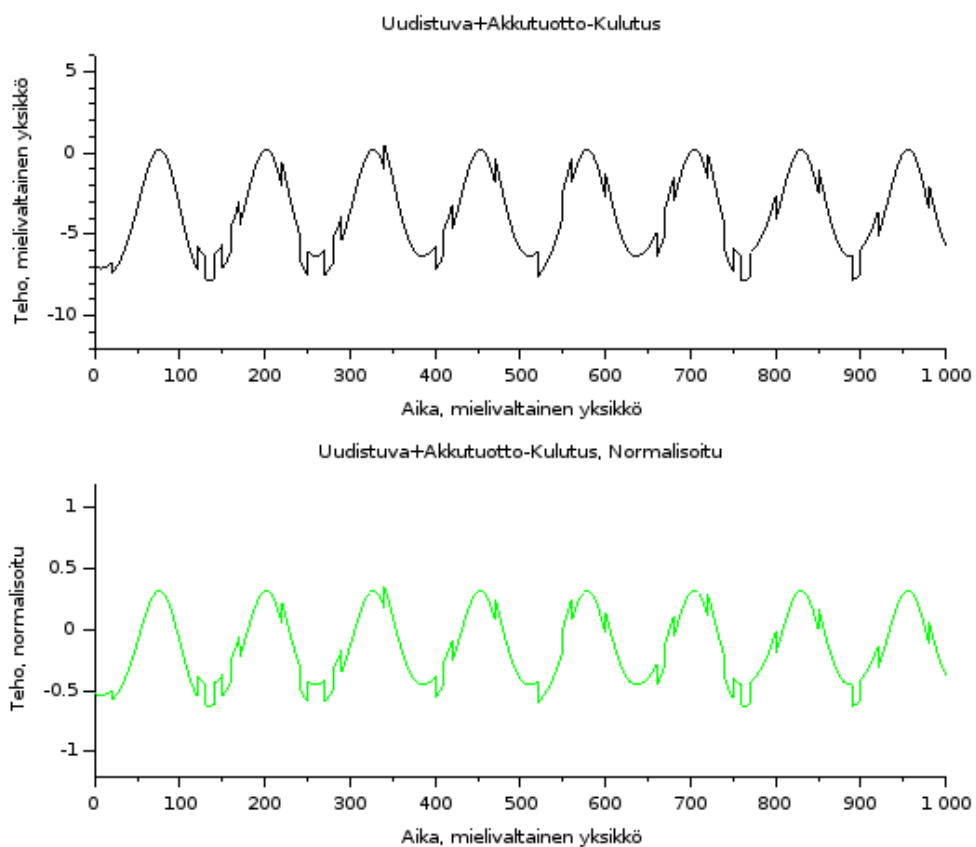
Kuvio 12. Pääverkon hallintasiignaalin kuvaaja.



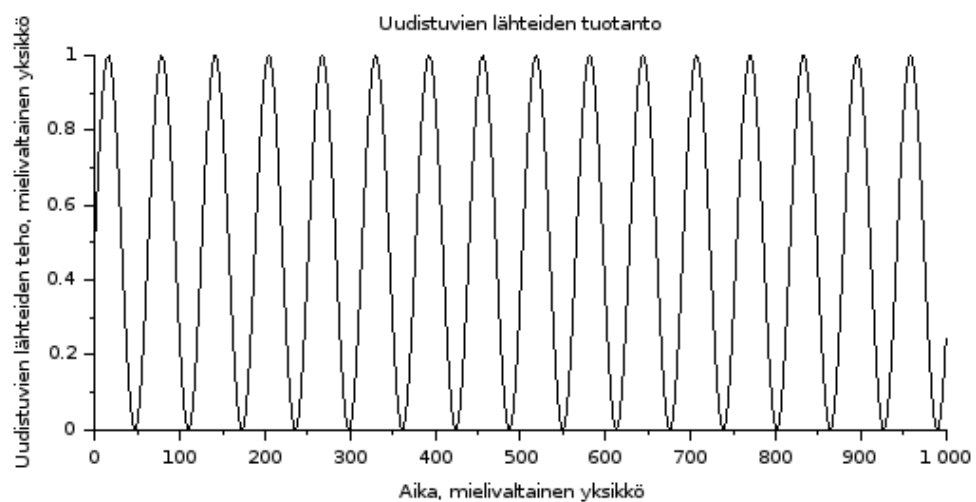
Kuvio 13. Generaattorin hallintasiignaalin kuvaaja.



Kuvio 14. Kokonaistuotannon ja kulutuksen erotuksen kuvaaja.



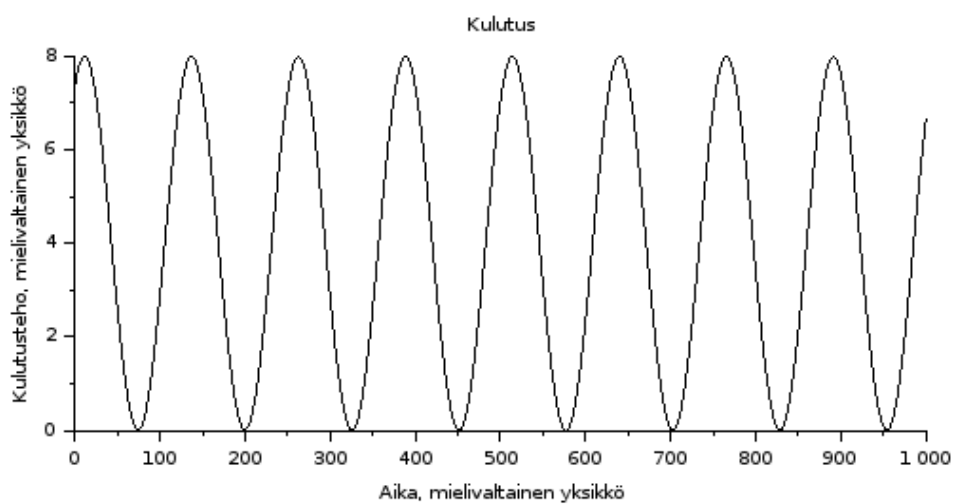
Kuvio 15. Uudistuvan ja akku tuotannon sekä kulutuksen erotuksen kuvaaja.



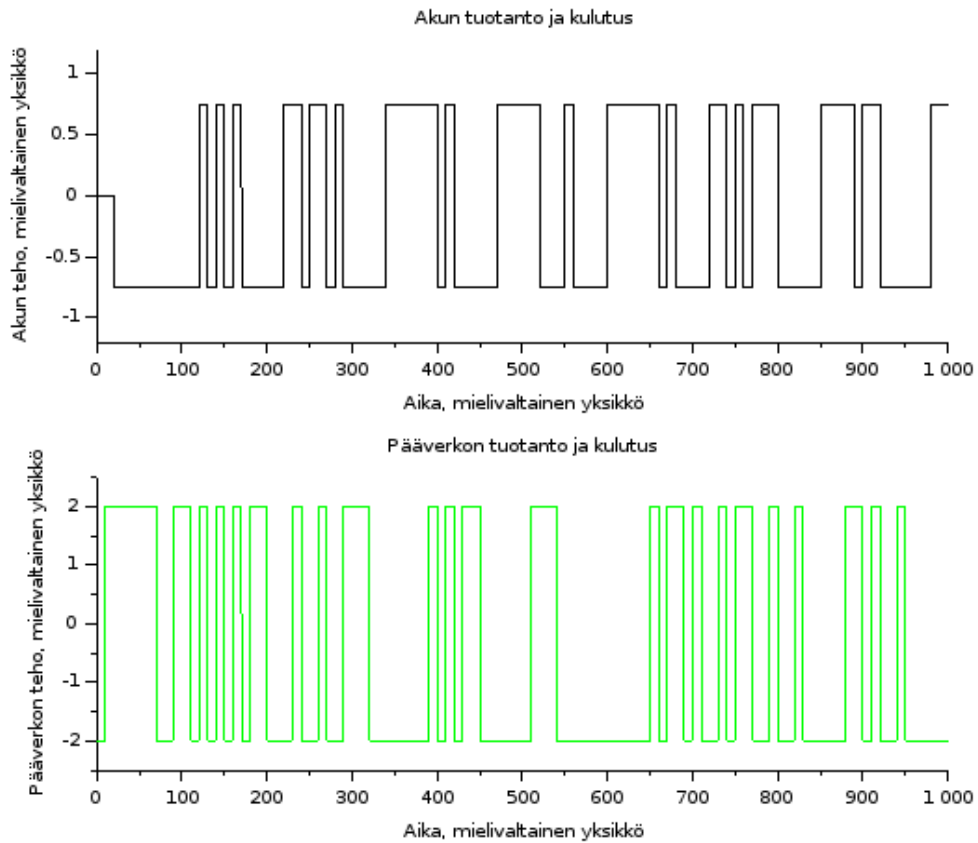
Kuvio 16. Uudistuvan tuotannon kuvaaja.



Kuvio 17. Generaattorin tuotannon kuvaaja.



Kuvio 18. Kulutuksen kuvaaja.



Kuvio 19. Akun ja pääverko tuotannon ja kulutuksen kuvaaja.

Tulokset ovat aikalailla odutuksia vastaavia. Generaattoria ja akkuja käytetään kulutuksen ollessa korkealla, akkuja ladataan kulutuksen ollessa matalalla, ylimääräistä energiaa myydään hinnan ollessa matalalla tai akkujen ollessa optimivarauksen yläpuolella ja akkujen varaus sekä tuotantojen ja kulutuksien erot pyörivät normalisoidun nollatason kohdalla.

YHTEENVETO

Sumea logiikkapiiri näyttää kuvaajien mukaan toimivan ainakin tässä testissä jotakuinkin odotusten mukaan.

Simulaatiossa olisi vielä parantamisen varaa. Simulaation alkuarvot ovat täysin mielenvaltaisia ja valittu lähinnä sen perusteella, että ne havainnollistavat logiikkapiirin toimintaa hyvin.

Lähdesignaalit on generoitu vakioilla, satunnaisesti tai sin-aalloilla. Ja mitään sähköteknisiä ominaisuuksia, kuten loistehoja, ei ole otettu mitenkään huomioon, vaan simulaatio kohtelee kaikkia signaaleja vain reaalilukuina jotka syötetään logiikkapiiriin.

Lisäksi akkusimulaattori on kehitetty mielivaltaisesti omasta päästä ja luultavasti vastaa oikean akun toimintaa vain siinä määrin että se toimivasti pitää kirjaa kuinka paljon mielivaltaisia energiayksiköitä se on varastoinut, ja onko se tyhjä tai täysi.

Nämä valinnat on tehty sen vuoksi että tämän laboratoriotyön tarkoituksena oli tutkia sumean logiikan käyttöä mikroverkon hallinnassa, ei varsinaisesti simuloida koko verkkoa. Näiden yksinkertaistukien avulla sai nopeasti mallinnettua jotakuinkin oikeaa verkkoa vastaavaa dynaamista syötettä, jolla logiikkapiirin käyttäytymistä voi tutkia.

LÄHDELUETTELO

S. D. Saranya, S. Sathyamoorthi & R. Gandhiraj. (2015). A Fuzzy Logic Based Energy Management System for a Microgrid. *ARPN Journal of Engineering and Applied Sciences, VOL. 10, NO. 6, APRIL 2015.*

Yu-Kai Chen, Yung-Chun Wu, Chau-Chung Song & Yu-Syun Chen. (2013). Design and Implementation of Energy Management System With Fuzzy Control for DC Microgrid Systems. *IEEE TRANSACTIONS ON POWER ELECTRONICS, VOL. 28., NO. 4, APRIL 2013.*

Sumea Logiikka (2017). Jarmo T. Alander. [Verkkodokumentti] Saatavissa: <http://lipas.uwasa.fi/~TAU/AUTO2050/slides.php?Mode=NoFrame&File=3000Fuzzy.txt&Page=-2&MicroExam=Off&Images=On&Menu=Off>

Stefano Leonori, Enrico De Santis, Antonello Rizzi and F.M.Frattale Mascioli. (2016). Multi Objective Optimization of a Fuzzy Logic Controller for Energy Management in Microgrids. *2016 IEEE Congress on Evolutionary Computation (CEC).*

About Microgrids (2017). Microgrids at Berkeley Lab. [Verkkodokumentti] Saatavissa: <https://building-microgrid.lbl.gov/about-microgrids>