

**Piikarbidipohjaisten
tehokomponenttien
säteilyvastemittauksissa käytetyn
suodatinpiirin mallintaminen**

Pro gradu - tutkielma, 27.11.2022

Tekijä:

ALEKSI KAUPPI

Ohjaaja:

ARTO JAVANAINEN



JYVÄSKYLÄN YLIOPISTO
FYSIKAN LAITOS

© 2023 Aleksi Kauppi

Julkaisu on tekijänoikeussäännösten alainen. Teosta voi lukea ja tulostaa henkilökohtaista käyttöä varten. Käyttö kaupallisiin tarkoituksiin on kielletty. This publication is copyrighted. You may download, display and print it for Your own personal use. Commercial use is prohibited.

Tiivistelmä

Kauppi, Aleksi

Piikarbidipohjaisten tehokomponenttien säteilyvastemittauksissa käytetyn suodatinpiirin mallintaminen

Pro gradu - tutkielma

Fysiikan laitos, Jyväskylän yliopisto, 2022, 62 sivua

Tässä työssä mallinnettiin piikarbidipohjaisten tehokomponenttien säteilynkestomittauksessa käytettyä ylipäästösuodatinpiiriä. GSI:n microbeamin säteilynkestomittauksissa estosuuntaiseen etujännitteeseen asetettuja piikarbidi- Schottky-diodeita ja MOSFETejä säteilytettiin 4.8 MeV/u Au-ioneilla, etujännitteiden ollessa välillä 150 - 300 V. Ioniosumista syntyneet nopeat virtapulsit erotettiin suuresta tasajännitteestä ylipäästösuodattimen avulla ja mitattiin piirin läpi jännitteenä. Samalla vuotovirtaa mitattiin ajan funktiona Keithleyn virtamittarilla. Itse säteilynkestomittaukset oli suoritettu ennen tätä työtä.

Tämän työn kokeellisessa osuudessa, mittauksessa käytetyn piirin taajuusvaste mitattiin ensin käyttäen Digilent Analog Discovery 2- signaaligeneraattoria/ oskilloskooppia. Piiristä luotiin sen jälkeen SPICE- malli, joka otti huomioon myös piirilevyn, koaksiaalikaapeleiden sekä oskilloskooppien impedanssit. Tarkat arvot parasiittisille impedansseille selvitetiin vertaamalla simuloituja taajuusvasteita mitattuihin Python-skriptin avulla.

Kun impedanssit tunnettiin taajuusvasteen avulla tarkasti, selvitetiin herätteen ja vasteen välinen yhteys ajan funktiona. Vertaamalla piirin läpi mitattujen jännitepulssien aikaleimoja mitatun vuotovirran muutoksien aikaleimoihin selvisi, että pysyvät portaittaiset nousut vuotovirrassa syntyvät todennäköisesti niistä virtapulsseista, joissa komponentin läpi kulkee suurempi kokonaisvaraus.

Avainsanat: piikarbidi, tehokomponentti, säteilynvaikutukset, SEE, SELC

Abstract

Kauppi, Aleksi

Modelling of filter circuit used in radiation degradation measurements of silicon carbide power components

Master's Thesis

Department of Physics, University of Jyväskylä, 2022, 62 pages.

In this work a high-pass circuit used in radiation hardness testing of silicon carbide power devices was modelled. Silicon carbide Schottky-diodes and MOSFETs were radiated by 4.8 MeV/u Au ions in GSI microbeam radiation hardness measurements. The components were reverse biased between 150-300 V. Quick current pulses induced by ion strikes were separated from the high DC voltages by high-pass filter and measured through the circuit as voltages. At the same time the leakage current was measured by Keithley current meter. The radiation measurements were carried out before this work.

In the experimental part of this work the frequency response of the used circuit was first measured using Digilent Analog Discovery 2 signal generator/oscilloscope. Then a SPICE model that included the impedances of the used coaxial cables, oscilloscopes and the circuit board, was created. Values for the parasitic impedances were found out by comparing the measured and simulated frequency responses by Python script.

After the impedances were found out, the connection between input current and output voltage as a function of time was defined. Comparing the time stamps of the measured voltage pulses to the time stamps of the increasing leakage current, it was found that the step-wise rises in the leakage current were probably caused by current pulses for which there is a greater total charge flowing through the device.

Keywords: silicon carbide, power device, radiation effects, SEE, SELC

Sisällys

Tiivistelmä	3
Abstract	5
1 Johdanto	9
2 Piikarbidi ja säteilyvuorovaikutukset	11
2.1 Piikarbidin ominaisuudet materiaalina	11
2.2 Säteilyn ja materian vuorovaikutukset	12
2.2.1 Lineaarinen energiansiirto	13
2.2.2 Varauksenkuljettajien luominen puolijohteissa	14
2.2.3 Yhden hiukkasen ilmiöt (Single Event Effects)	14
2.2.4 Piikarbidi ja säteily	15
3 Kokeelliset menetelmät	17
3.1 Säteilynkestomittausten esittely	17
3.2 Käytetyn piirin taajuusvasteen mittaaminen	19
3.3 Piirin taajuusvasteen mallintaminen LTspicella	20
3.4 Piirin aikakäytöksen mallintaminen	26
4 Tulosten tarkastelu	29
4.1 Säteilynkestomittausten visualisointi	29
4.2 Säteilynkestomittausten analysointi mallin avulla	31
4.3 Vuotovirran kasvu ja suuremman varauksen pulssit	34
5 Johtopäätökset	37
Lähteet	37
A Parhaan simulaation haarukointi	41
B Simulaatioiden eksponenttisoitus	45

C Luokat datan muotoiluun	49
D Mittausten sovitus ja kerääminen samaan tiedostoon	53
E Vuotovirran nousujen ja suurempien jännitepulssien aikojen vertailu	57
F Vuotovirran ja sen derivaatan kuvaajat	61

1 Johdanto

Avaruudessa on tuhansia satelliitteja ja luotaimia jotka sisältävät valtavat määrät erilaista elektroniikkaa[1]. Käytetyn elektroniikan on oltava erityisen kestävä ja luotettava. Elektroniikan on kestävä avaruudessa muun muassa suurilla lämpötilanvaihtelulla, sähkömagneettisilla häiriöillä, mekaanisella rasituksella, sekä matalilla paineilla. Koska satelliitit kiertävät maata ilmakehän suojaavan vaikutuksen ulkopuolella, täytyy niiden kestää myös kosmista säteilyä. [2]

Kosminen säteily, jolle maata kiertävät satelliitit altistuvat, on peräisin suurimmaksi osaksi auringosta. Se sisältää elektroneja, protoneja, alfa-hiukkasia sekä myös raskaita ioneja. Auringosta tulevan säteilyn määrä vaihtelee auringon 11 vuoden sykleissä muuttuvan aktiivisuuden mukaan. Auringon ollessa aktiivisimmillaan esiintyy paljon aurinkopilkkuja, auringonpurkauksia ja koronan massapurkauksia, jotka kaikki lisäävät auringosta peräisin olevien hiukkasten määrää. Satelliitit voivat myös altistua muualta galaksista tai galaksin ulkopuolelta peräisin oleville suurenergisille hiukkasille, jotka ovat peräisin mm. tähdistä, galaksien ytimistä ja supernovista. [3]

Maan magneettikentän takia auringosta ja muualta avaruudesta peräisin olevat varatut hiukkaset voivat jäädä loukkuun maata ympäröivään avaruuteen, muodostaen runsaasti säteilyä sisältäviä vyöhykkeitä, joita kutsutaan Van Allenin vyöhykkeiksi. Sisempi Van Allenin vyöhykkeistä sisältää pääasiassa protoneja ja ulompi elektroneja. [4]

Elektroniikan vastetta eri tyyppiseen säteilyyn voidaan testata hiukkaskiihdyttimien avulla. Tämän tutkielman aiheena on säteilynkestomittauksessa käytetty ylipäästösuodinpiiri. Mittauksessa piikarbidipohjaisia tehokomponentteja säteilytetään kulta-ioneilla, komponenttien ollessa suuressa etujännitteessä. Komponenttien vuotovirtaa ja ioni-osumista syntyneitä virtapulseja mitattiin säteilytyksen aikana. Virtapulssit erotettiin DC-etujännitteestä ylipäästösuodinpiirin avulla ja mitattiin piirin läpi jännitteenä.

Tutkielmassa esitellään ensin piikarbidin ominaisuudet puolijohteena, jonka jälkeen avataan materiaalin ja säteilyn vuorovaikutuksiin liittyviä käsitteitä. Sen jälkeen esitellään GSI:llä tehtyjä mittauksia ja mittauseräilyä, joissa tutkielman aiheena

oleva piiri oli käytössä. Huomattakoon, että mittaukset oli jo suoritettu ennen tämän työn aloittamista.

Tämän työn kokeellisessa osuudessa käytetyn ylipäästösuodinpiirin taajuusvaste mitataan ja piiristä luodaan parasiittiset impedanssit huomioon ottava malli. Mitatun taajuusvasteen ja luodun mallin avulla selvitetään LTspicea ja Pythonia käyttäen tuntemattomat impedanssit. Kun impedanssit tunnetaan voidaan selvittää piirin siirtofunktio. Siirtofunktion avulla saadaan yhteys mitattujen jännitepulssien ja komponentissa tapahtuneiden virtapulssien suuruuden välille. Pulssien aikaleimoja voidaan sen jälkeen verrata komponentin vuotovirran aikaleimoihin.

2 Piikarbidi ja säteilyvuorovaikutukset

Tässä luvussa tutustutaan piikarbidin ominaisuuksiin materiaalina, sekä esitellään työn kannalta oleellisia käsitteitä, kuten lineaarinen energiansiirto (linear energy transfer, LET) ja varauksenkuljettajien luominen puolijohdeissa. Lisäksi tutustutaan yksittäisten hiukkasten aiheuttamiin ilmiöihin (single event effects, SEE) puolijohdeissa.

2.1 Piikarbidin ominaisuudet materiaalina

Piikarbidi (SiC) on piistä ja hiilestä muodostunut IV-IV- tyyppin puolijohde. Se on materiaalina kestävä, kevyttä ja kovaa. Lisäksi piikarbidi johtaa ja kestää hyvin lämpöä [5]. Elektroniikassa piikarbidin suurimmat edut ovat kuitenkin sen fysikaaliset ominaisuudet puolijohdeena.

Merkittävin fysikaalinen ero puolijohdekomponenttien toiminnan kannalta piikarbidin ja piin välillä on energia-aukon suuruus. Piikarbidin energia-aukko on 2.36 – 3.26 eV riippuen polytyypistä, eli materiaalin kiderakenteesta [6]. Piin energia-aukko taas on vain 1.12eV [6]. Koska piikarbidin energia-aukko on suurempi, elektronit tarvitsevat enemmän energiaa päästäkseen valenssivyöltä johtavuusvyölle.

Suuremmasta energia-aukosta johtuen piikarbidi kestää lähes kymmenen kertaa suurempia sähkökenttiä ilman läpilyöntiä. Piin sähkökentän läpilyöntivoimakkuus on vain 0.25MV/cm, kun taas piikarbidin sähkökentän läpilyöntivoimakkuus on jopa 2.2MV/cm [6]. Koska piikarbidi kestää suuria sähkökenttiä, kestää se myös paljon suurempia jännitteitä laitteissa, mahdollistaen komponentit jopa yli 10kV jännitteille.

Sähkökenttien keston ansiosta epitaksiaalikerrokset voidaan valmistaa ohuemmiksi kuin piikomponenteissa ja siten piikarbidikomponenteista voidaan tehdä paljon pienempiä. Pienempi koko vähentää loiskapanssianssia ja mahdollistaa siten nopeammat kytkentätaajuudet.

Koska läpilyöntivoimakkuus on suurempi, voidaan myös seostusta kasvattaa. Seostuksen kasvatus pienentää ON-vastusta. ON-vastus tarkoittaa lähteen ja nielun välistä vastusta komponentin ollessa johtavassa tilassa. Mitä pienempi ON-vastus, sitä

pienempi tehonkulutus. ON-vastus on piikarbidikomponenteissa korkeilla jännitteillä jopa satoja kertoja pienempi kuin piikomponenteissa. [7]

Pienestä ON-vastuksesta johtuen ei nopeutta vaativiin komponentteihin tarvita vähemmistövarauksenkuljettaja- rakenteita, kuten IGBT (insulated-gate bipolar transistor) tai tyristori, vaan komponenteista voidaan tehdä enemmistövarauksenkuljettaja-komponentteja, kuten MOSFET ja Schottky-diodi. Enemmistövarauksenkuljettaja-komponentit ovat nopeampia kuin vähemmistö-varauksenkuljettaja-komponentit. Siksi piikarbidikomponentit sopivat korkeiden jännitteiden ja nopeiden kytkentätaajuuksien sovelluksiin. [7] [8]

Piikarbidin liikkuvuus (mobility) on pienempi kuin piillä, mutta elektronin ajelehtimisnopeus saturoituu myöhemmin, mahdollistaen nopeamman kytkentätaajuuden. Koska liikkuvuus on pienempi piikarbidikomponentit eivät kuitenkaan sovellu erittäin matalien jännitteiden sovelluksiin. [6]

Suuresta energia-aukosta johtuen piikarbidikomponentit kestävät korkeampia käyttölämpötiloja ($T > 300^{\circ}\text{C}$) kuin piikomponentit [6]. Lisäksi piikarbidin sulamispiste on niin korkea että faasitransitioita ei tapahdu käytössä. Vaikka piikarbidi kestääkin suuria lämpötiloja, kotelot ja kondensaattorit eivät välttämättä kestä sellaisia lämpötiloja, joilla piikarbidikomponentti toimii.

Piikarbidin lämmönjohtavuus on suurempi kuin metalleilla, joten käytössä syntynyt lämpö siirtyy tehokkaasti pois. Koska piikarbidikomponentit ovat pieniä, ovat ne toisaalta herkempiä lämpötilan vaihteluille. [6]

Suurten tehojen ja korkeiden taajuuksien sovelluksissa sähkömagneettisilta interferensseiltä suojautuminen vaatii vielä kehitystyötä. Myöskään piikarbidikomponenttien yleinen käyttöluotettavuus ei ole vielä aivan samalla tasolla piikomponenttien kanssa. [9]

2.2 Säteilyn ja materian vuorovaikutukset

Säteilyn kulkiessa materiassa se vuorovaikuttaa väliaineen atomien kanssa. Säteily voi olla tyypiltään sähkömagneettista säteilyä, energettisiä elektroneja, neutroneja, protoneja tai ioneja [10]. Tässä tutkielmassa keskitytään erityisesti energettisten raskaiden ionien, kuten Au, vaikutuksiin piikarbidissa.

Kulkiessaan väliaineen läpi, positiivisesti varattu ioni vuorovaikuttaa aineen atomien kanssa Coulombin voiman kautta. Ioni voi virittää väliaineen atomin elektroneja korkeammille energiatiloille valenssivyöltä johtavuusvyölle (eksitaatio), tai irroittaa

valenssielektronin, jolloin atomi ionisoituu. Ionien tapauksessa suurin osa vuorovaikutuksista koostuu eksitaatioista ja ionisaatioista. Ioni luovuttaa liike-energiaansa vuorovaikutusten kautta väliaineen atomeihin, jolloin hiukkasen nopeus pienenee. [11]

Hiukkanen voi myös vuorovaikuttaa yksittäisen elektronin kanssa siten, että elektroni saa irrotessaan myös huomattavan määrän liike-energiaa. Suuren liike-energian elektroneita kutsutaan delta-elektroneiksi. Delta-elektronit voivat myös itsessään vuorovaikuttaa väliaineen atomien kanssa ja eksitoida tai ionisoida atomeja. [10]

Jos ionin energia on riittävän suuri, se voi myös sirota atomin ytimestä, jolloin osan energiasta siirtyy atomin liike-energiaksi. Atomin saadessa tarpeeksi liike-energiaa, se voi myös irrota hilasta, mikäli energia ylittää atomien välisen sidosenergian. Riittävän suurilla energioilla voi tapahtua myös ydinreaktioita, joiden reaktiotuotteet kulkevat aineessa aiheuttaen uusia vuorovaikutuksia. [10]

2.2.1 Lineaarinen energiansiirto

Lineaarinen energiansiirto (Linear energy transfer) tarkoittaa energiaa, jonka energettinen hiukkanen siirtää väliaineeseen kuljettua matkaa kohden. Se määritellään

$$L = \frac{dE}{dl} , \quad (1)$$

missä dE on materiaaliin siirtynyt energia, ja dl on hiukkasen materiaalissa kulkema matka. Energiansiirron määrä riippuu sekä säteilyn tyypistä että materiaalista.

Rajoittamaton energiansiirto (unrestricted linear energy transfer) vastaa suuruudeltaan pysäytysvoimaa (stopping power). Pysäytysvoima tarkoittaa sitä, kuinka paljon energiaa materiassa kulkeva hiukkanen menettää kuljettua matkaa kohden.

Energiansiirto rajoittuu yleensä niin, että se sisältää vain tietyn energian alittavat vuorovaikutukset. Rajoitettu lineaarinen energiansiirto vastaa väliaineen lokaalisti vastaanottaman energian määrää. Se ei siten sisällä suuren liike-energian delta-elektroneja, jotka vuorovaikuttavat kaukana hiukkasen kulkemasta radasta. Linearisesta energiansiirrosta voidaankin puhua myös rajoitettuna pysäytysvoimana. [12]

2.2.2 Varauksenkuljettajien luominen puolijohteissa

Termisessä tasapainossa puolijohteen elektronit ovat jakautuneet valenssi- ja johtavuusvöiden välille Fermi-Dirac- jakaumafunktion mukaan. Valenssivyön elektronin saadessa riittävästi energiaa esimerkiksi säteilyn vaikutuksesta, se voi virittyä johtavuusvyölle. Tällöin elektroni jättää valenssivyölle aukon ja syntyy elektroni-aukko-pari. Keskimääräinen elektroni-aukko-parin syntymiseen tarvittava energia on 4H-SiC:lla $\epsilon = 7.28\text{eV}$ [13].

Johtavuusvyöllä olevat elektronit ja valenssivyöllä olevat aukot voivat molemmat kuljettaa varausta ja siten virtaa atomihilan läpi. Mikäli säteilytetty komponentti on asetettu jännitteeseen, elektronit ja aukot alkavat liikkua sähkökentässä ja aiheuttavat virtapulssin.

Hilassa liikkuvat varauksenkuljettajat voivat myös virittää hila-atomien elektroneja ja synnyttää lisää varauksenkuljettajapareja, jolloin syntyy lisää virtaa. Tätä kutsutaan lumivyöryefektiksi.

2.2.3 Yhden hiukkasen ilmiöt (Single Event Effects)

Mikäli ionin materiaaliin luovuttama energia on riittävä tuottamaan elektroni-aukko-pareja ja mikäli komponentti on riittävän suuressa jännitteessä, yksittäisestä ioniosumasta synnyntynyt virtapulssi voi tuhota komponentin tai muuttaa komponentin käytöstä. Seuraavassa listataan muutamia tyypillisiä tehokomponenteissa havaittavia yhden hiukkasen ilmiöitä.

SEB Single event burnout Ionin osuessa komponentin epitaksiaalikerroksen ja substraatin rajapintaan, syntyy elektroni-aukkopareja. Mikäli LET on riittävän suuri, ioniosuma synnyttää niin paljon varauksenkuljettajia, että komponenttiin syntyy ei-toivottu matalan vastuksen virtaa johtava polku. Polku muodostuu MOSFETeilla nielun ja lähteen välille ja diodeilla anodin ja katodin välille. Kun komponentti on suuressa jännitteessä varauksenkuljettajat liikkuvat sähkökentässä ja syntyy nopea virtapiikki. Suuri virta tuottaa lämpöä, joka hajottaa komponentin. Tätä ilmiötä kutsutaan single-event burnoutiksi. [14] [15]

SEGR single event gate rupture Single event gate rupture tarkoittaa ilmiötä jossa ionin osuma oksidikerrokseen aiheuttaa virtaa vuotavan polun hilan eristeeseen. Hilan läpi kulkeva ioni heikentää hilaa, sekä kasvattaa hilan sähkökenttää joka mahdollistaa tuhoisan virtapiikin. Virta voi muuttaa komponentin käytöstä tai tuhota sen kokonaan. [16]

SELC single event leakage current Single event leakage current on ilmiö joka voi tapahtua jo pienemmilläkin jännitteillä. Ionin osuessa komponenttiin, komponentin vuotovirta (MOSFETEilla nielusta lähteeseen, diodeilla estosuuntainen) voi kasvaa pysyvästi. Kasvanut vuotovirta ei välttämättä hajota komponenttia kokonaan, mutta voi muuttaa sen toimintaa. Vuotovirran suuruus riippuu materiaalista, lineaarisesta energiansiirrosta sekä etujännitteestä. Kun vuotovirta kasvaa riittävän suureksi voi tapahtua SEB. SELC on tyyppillisempi piikarbidikomponenteille, piikomponenteillakin vastaavanlainen ilmiö on kuitenkin havaittu. [17]

2.2.4 Piikarbidi ja säteily

Piikarbididiodien säteilyvaste voidaan jakaa kolmeen eri alueeseen. Matalilla jännitteillä säteilyn aiheuttama ionisaatio kasvattaa vuotovirtaa, joka ei kuitenkaan muuta komponentin toimintaa pysyvästi vaan häviää säteilyn loppuessa. Korkeammille jännitteille siirryttäessä vuotovirta alkaa kasvaa pysyvästi ja muuttaa komponentin käytöstä (SELC). Kun jännite säteilytyksen alla nousee riittävän suureksi, komponentti tuhoutuu jo yksittäisestä hiukkasosumasta (SEB). SEB:tä ilmenee säteilytyksen alaisena jo reilusti nimellisjännitteiden alapuolella. Säteilyvaste riippuu jännitteen lisäksi ionin LET:stä ja kokonaisvuosta. [16]

3 Kokeelliset menetelmät

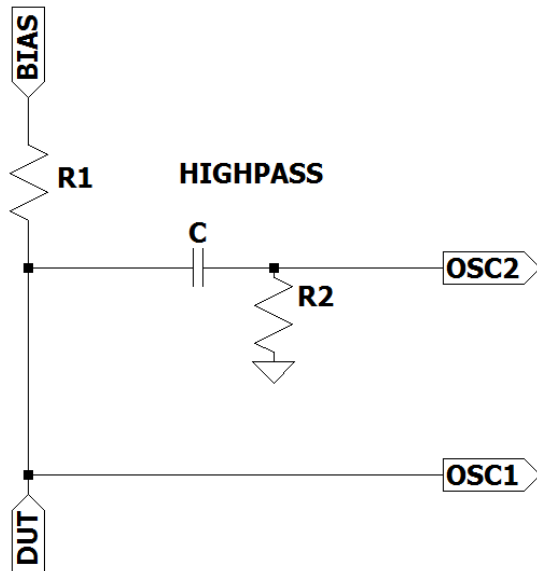
Tässä luvussa esitellään ensin muiden toimesta tehdyt säteilynkestomittaukset, joiden tuloksia tässä työssä analysoidaan. Sen jälkeen käsitellään tämän työn kokeellinen osuus eli säteilynkestomittauksissa käytetyn piirin taajuusvasteen mittaaminen ja siinä käytetyt välineet, sekä piirin simulointi ja siinä käytetyt ohjelmistot.

3.1 Säteilynkestomittausten esittely

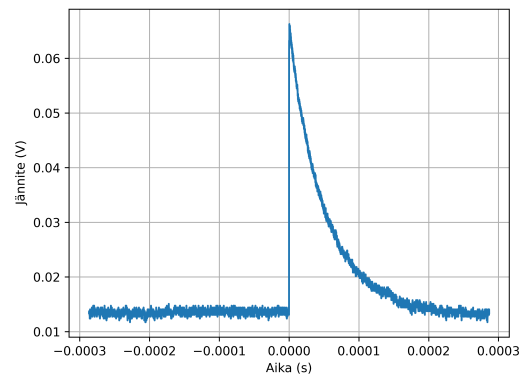
GSI:n kohdistetulla ionisuihkulla mitatuissa säteilynkestomittauksissa estosuuntaiseen etujännitteeseen asetettuja piikarbidi- Schottky-diodeja sekä MOSFETeja säteilytettiin 4.8 MeV/u Au-ioneilla. Komponenttien etujännitteet olivat välillä 150 - 300 V.

Osumista syntyneet virtapulssit erotettiin suuresta DC-etujännitteestä RC- ylipäästösuodattimen avulla. Pulssit mitattiin suodattimen läpi jännitteinä Digilentin Analog Discovery 2:lla. Samalla mitattiin komponentin vuotovirtaa ajan funktiona Keithleyn virtamittarilla. Mittausasetelma on esitetty kuvassa 1. DUT (Device under test) on säteilytettävä komponentti, BIAS on käytetty etujännite, OSC2 mittaa jännitepulsseja ja OSC1 vuotovirtaa. Yksittäinen mitattu vastejännitepulssi on esitetty kuvassa 2 ja kaikki mittauksen MOSFET $V_{ds} = 300V$ vastejännitepulssit on esitetty histogrammina kuvassa 3. Säteilytyksen aikana mitattu vuotovirta on esitetty kuvassa 4.

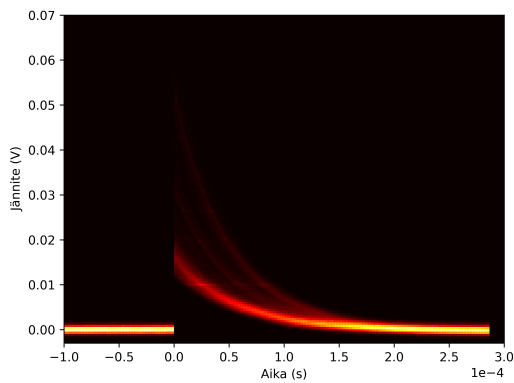
Tässä työssä on tarkoitus selvittää, miten mittauksissa käytetty suodatinpiiri reagoi erilaisiin herätteisiin ja siten selvittää, millaiset herätevirtapulssit aiheuttavat mitatut vastejännitepulssit. Kun herätevirtapulssien ja vastejännitepulssien välinen yhteys tunnetaan, voidaan myös tarkastella millaisia vuotovirran nousujen lähistöllä tapahtuvat ioniosumat ovat luovutetun varauksen suhteen.



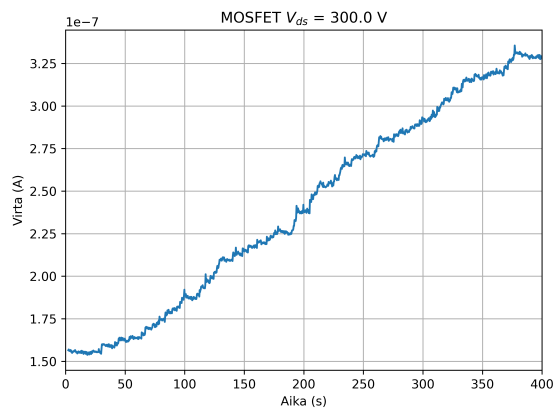
Kuva 1. Mittausasetelma



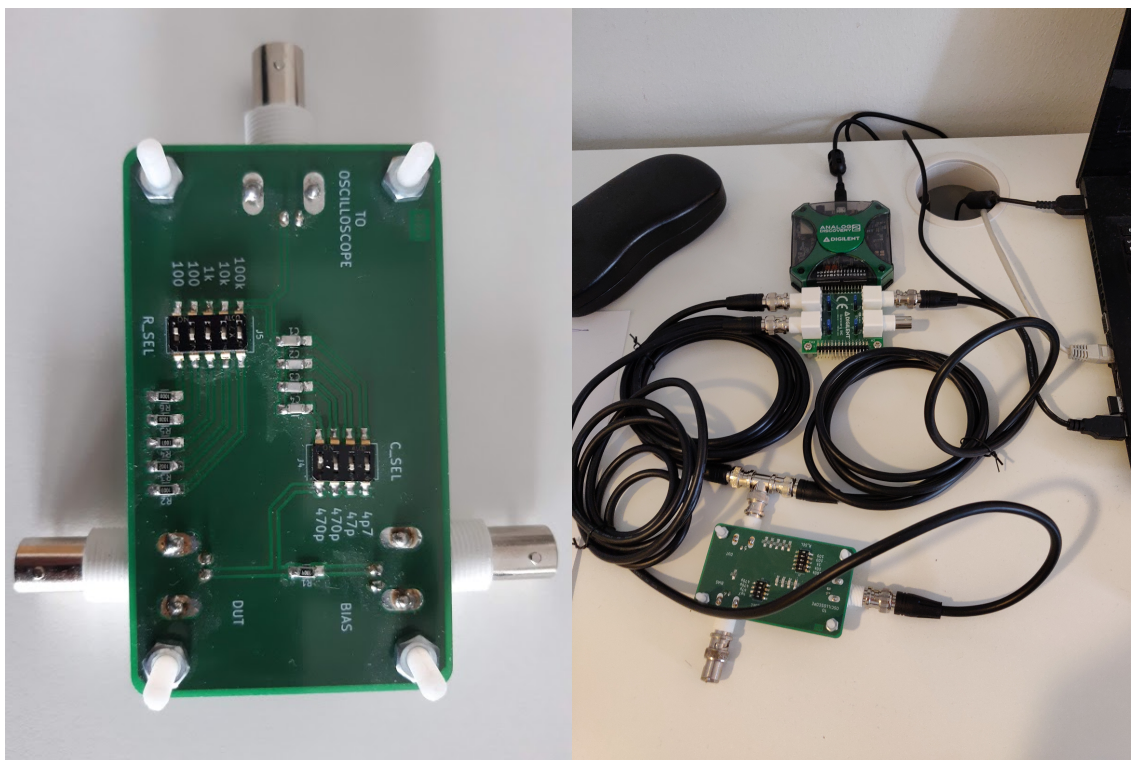
Kuva 2. Yksittäistä hiukkasosumaa vastaava mitattu vastejännitepulssi



Kuva 3. 2D-Histogrammi kaikista mitatuista vastejännitepulseista MOSFETin etujännitteellä $V_{ds} = 300V$



Kuva 4. Vuotovirta ajan funktiona MOSFET $V_{ds} = 300V$



Kuva 5. Tarkasteltu ylipäästösuodatin

Kuva 6. Taajuusvasteen mittausasetelma

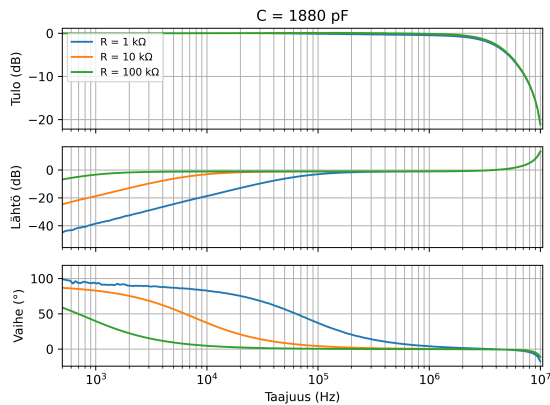
3.2 Käytetyn piirin taajuusvasteen mittaaminen

Jotta suodatinpiirin käytös saadaan selville, on tunnettava sen taajuusvaste, eli vahvistus ja vaihe. Piirin taajuusvastetta mitattiin tuottamalla Digilentin Analog Discovery 2:n funktiogeneraattorilla 1 voltin siniaaltopyyhkäisy taajuuksilla 100Hz – 10MHz. Analog Discovery 2:n oskilloskoopeilla mitattiin sekä funktiogeneraattorin tuottama heräte että ulostulovaste. Mittausasetelma ja suodatinpiiri on esitetty kuvissa 5 ja 6. Käytetyt välineet on esitetty taulukossa 1. Taajuusvaste mitattiin kondensaatto-

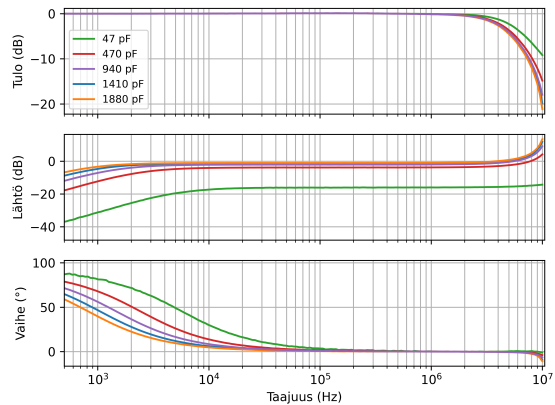
Taulukko 1. Käytetyt välineet

PC
Digilent Waveforms- ohjelmisto
Digilent Analog Discovery 2 funktiogeneraattori ja oskilloskooppi.
Ylipäästösuodatinpiiri säädettävillä kapasitanssin ja vastuksen arvoilla 3 kpl 2 metrin RG58C/U 50 Ω koaksiaalikaapelia

rin ja vastuksen arvojen yhdistelmille 4.7pF – 1880pF ja 100Ω – 100kΩ. Vastuksen vaikutus taajuusvasteeseen on esitetty kuvassa 7 ja kapasitanssin vaikutus kuvas-



Kuva 7. Piirin taajuusvaste vastuksen arvoilla $R = 1\text{k}\Omega - 100\text{k}\Omega$ ja vakiokapasitanssilla $C = 1880\text{pF}$



Kuva 8. Piirin taajuusvaste kapasitanssin arvoilla $C = 47\text{pF} - 1880\text{pF}$ ja vakiovastuksella $R = 100\text{k}\Omega$

sa 8. Suuremmilla vastuksilla ja kapasitansseilla suodattimen rajataajuus (cut-off frequency) pienenee yhtälön

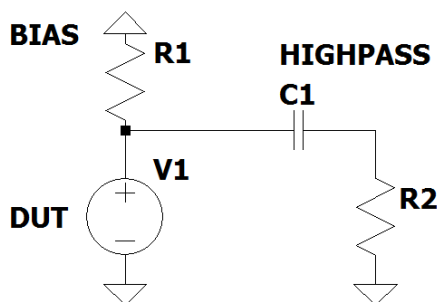
$$f_c = \frac{1}{2\pi RC} \quad (2)$$

mukaan, missä R on vastus ja C kapasitanssi. Sisääntulojännitteen amplitudi pienenee korkeilla taajuuksilla funktiogeneraattorin rajoituksista johtuen, mutta koska ulostulovahvistus on suhteutettu sisääntuloon, saadaan ulostulon vahvistus silti mitattua oikein. Toinen funktiogeneraattorin rajoitus on se, että sillä ei pysty tuottamaan yli 10 MHz taajuuksia, minkä takia piirin resonanssitaajuus jää juuri mittausalueen ulkopuolelle. Huomataan myös, että komponenttien biasoimiseen käytetty etujännite ei vaikuta taajuusvasteeseen, koska se on tasajännitettä, joten se voidaan mittauksessa maadoittaa.

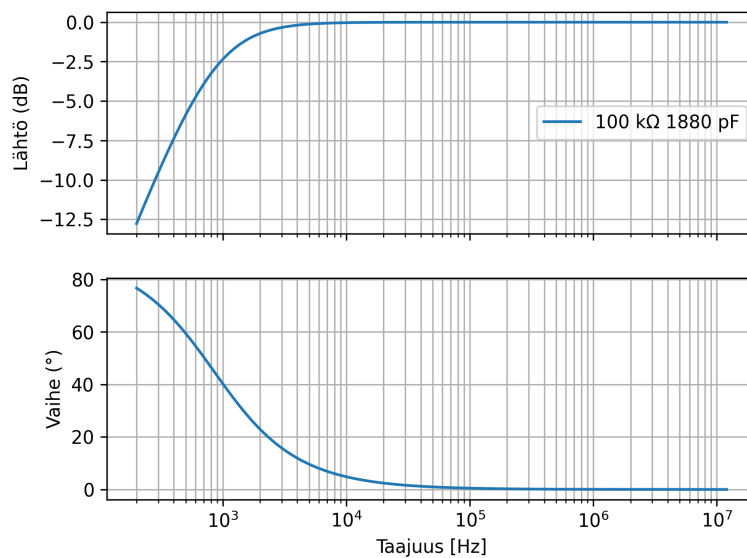
3.3 Piirin taajuusvasteen mallintaminen LTspicella

Piirin taajuusvastetta voidaan mallintaa luomalla piiristä LTspice-malli. LTspice on SPICEen (Simulation Program with Integrated Circuit Emphasis) perustuva analogielektronikan piirisimulointiohjelma. LTspicella piirrettiin ensin mittausta vastaava kytkentäkaavio. Sen jälkeen piirin käytöstä simuloitiin erilaisilla herätteillä. Mittausasetelman simuloimiseksi jännitelähteellä tuotettiin 1 voltin siniaaltopyyhkäisy ja simuloitiin piirin ulostulon taajuusvaste.

Ideaalinen RC-ylipäästösuodatinpiiri päästää vastuksen ja kondensaattorin arvojen mukaisesti kaikki korkeiden taajuuksien AC-signaalit piirin läpi. Kuten taa-



Kuva 9. Ideaalinen piiri



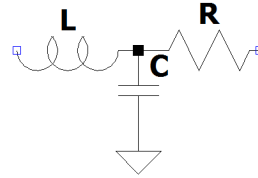
Kuva 10. Ideaalinen taajuusvaste

juusvasteen mittauksessa, DC-jännite ei vaikuta piirin toimintaan, joten se voidaan korvata maadoituksella. Ideaalinen RC-piiri ja simuloitu taajuusvaste on esitetty kuvissa 9 ja 10.

Jotta voidaan tarkemmin mallintaa todellista mitattua taajuusvastetta, on otettava huomioon myös koaksiaalikaapelien, sekä oskilloskooppien impedanssit. Oskilloskooppien impedanssia voidaan mallintaa vastuksen ja kondensaattorin rinnankytkennällä (kuva 11). Oskilloskooppien sisääntuloimpedanssi on valmistajan mukaan rinnankytkettynä $R = 1\text{M}\Omega$ ja $C = 24\text{pF}$. Koaksiaalikaapeleita voidaan mallintaa sarjaan kytketyillä vastuksella ja kelalla, sekä rinnankytketyillä kondensaattorilla. Malli on esitetty kuvassa 12.

OSCILLOSCOPE

Kuva 11. Oskilloskoopin sisääntulon ekvivalenttipiiri

COAX

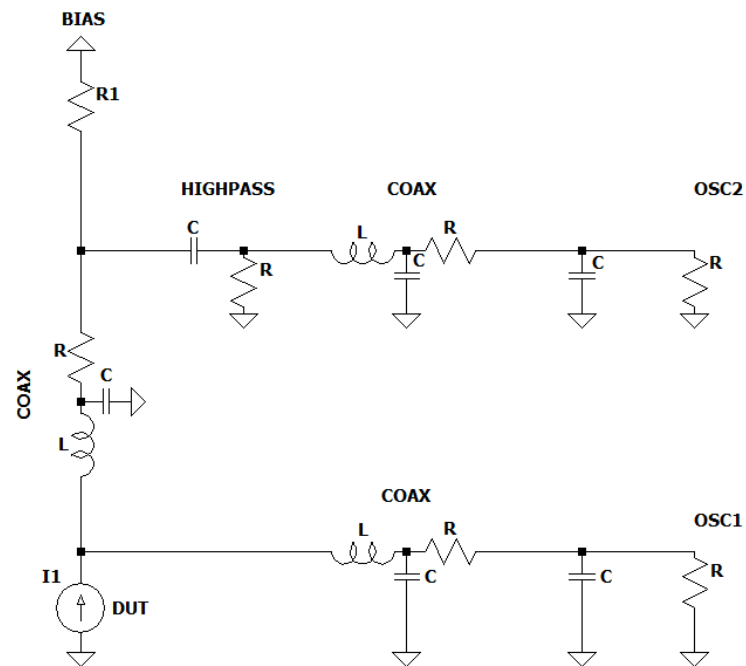
Kuva 12. Koaksiaalikaapelin ekvivalenttipiiri

Mittauksessa käytettyjen koaksiaalikaapeleiden nimellisarvo kapasitanssille on $C = 101\text{pF/m}$ ja induktanssille $L = 0.077\mu\text{H/ft}$, joista saadaan 3 metrin koaksiaalikaapeleille arvot $C = 202\text{pF}$ ja $L = 0.5053\mu\text{H}$. Vastukseksi valittiin sopivan pieni arvo $R = 50\text{m}\Omega$. Kun piiriin lisätään mallit oskilloskoopeista ja koaksiaalikaapeleista (kuva 13), taajuusvasteen muoto on jo oikeanlainen, mutta resonanssitaajuus ei kuitenkaan vastaa mittauksia (kuva 14). Syy tähän on se, että piirilevyn geometriasta johtuvia parasiittisia impedansseja ei tunneta. Lisäksi koaksiaalikaapeleiden todelliset impedanssit voivat poiketa nimellisarvoista. Piirilevystä johtuvat impedanssit voidaan kuitenkin sisällyttää koaksiaalikaapeleiden malleihin.

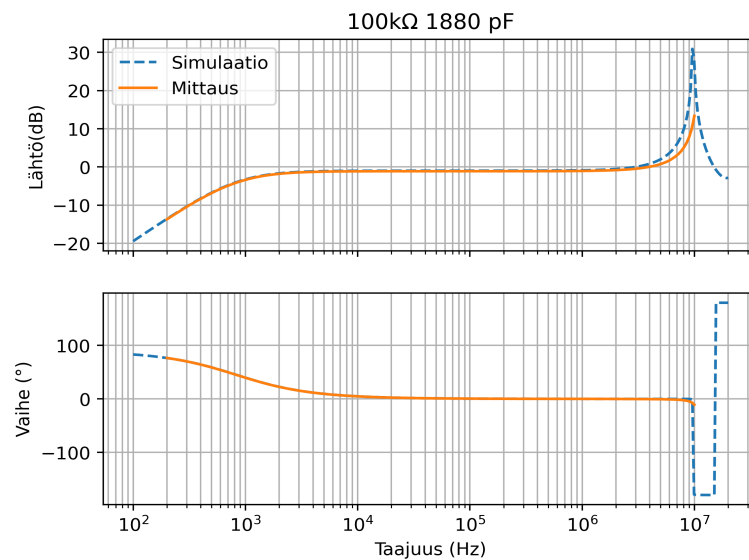
Koaksiaalikaapeleiden ja oskilloskooppien mallien komponenttien arvoja ei tunneta tarkkaan. Simuloimalla taajuusvasteita eri komponenttien arvoilla, voidaan kuitenkin selvittää, millä komponenttien arvoilla mallit ovat lähimpänä mittausta. Oskilloskooppien mallin arvojen muuttamisella ei ollut merkittävää vaikutusta taajuusvasteeseen. Koaksiaalikaapeleiden mallin kapasitanssin ja induktanssin arvoja muuttamalla taajuusvaste kuitenkin muuttui.

Tuntemattomien impedanssien selvittämiseksi LTspicellä simuloitiin ensin taajuusvasteita eri koaksiaalikaapeleiden kapasitanssien ja induktanssien arvoilla ja verrattiin taajuusvasteita mitattuihin. Sekä induktanssin että kapasitanssin kasvattaminen siirsi resonanssipeikkiä matalammille taajuuksille. Monenlaisilla kapasitanssi-induktanssi-pareilla sai siksi hyvin samankaltaisen taajuusvasteen. Huomattiin kuitenkin, että vain koaksiaalikaapelin kapasitanssin muutos vaikuttaa taajuusvasteeseen matalammilla, alle 500 kHz taajuuksilla, mutta induktanssin muutos ei.

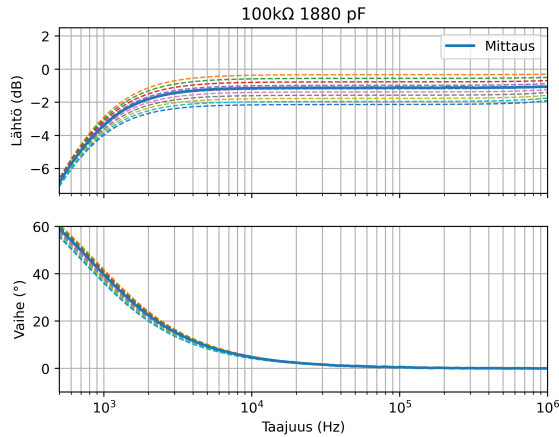
Parhaiden arvojen selvittämiseksi kirjoitettiin python-skripti, jonka avulla saatiin haarukoitua simuloituista taajuusvasteista, se joka on lähimpänä mittausta. Skripti on esitetty liitteissä. Ensin simuloitiin piirin taajuusvastetta eri kapasitanssin



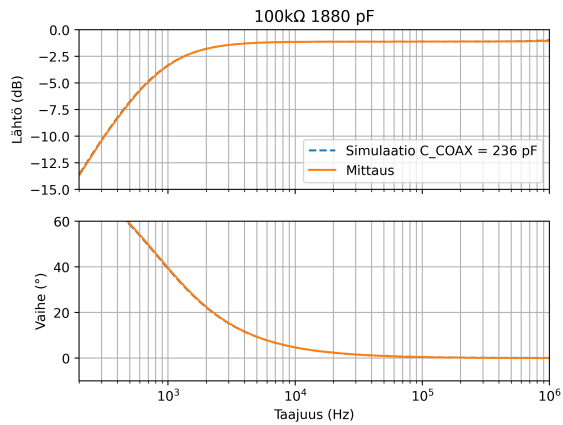
Kuva 13. SPICE-piiri mallista, joka ottaa huomioon oskilloskooppien ja koaksiaalikaapeleiden impedanssit



Kuva 14. SPICE- mallin simuloitu taajuusvaste ja mitattu taajuusvaste



Kuva 15. Piirin simuloitua taajuusvasteita (katkoviivat) eri koaksiaalikaapelin kapasitanssin arvoilla. Kapasitanssin kasvaessa vahvistus pienenee.



Kuva 16. Python-skriptillä haarukoitua simuloitua taajuusvasteita, joka on lähimpänä mitattua taajuusvastetta matalilla taajuuksilla.

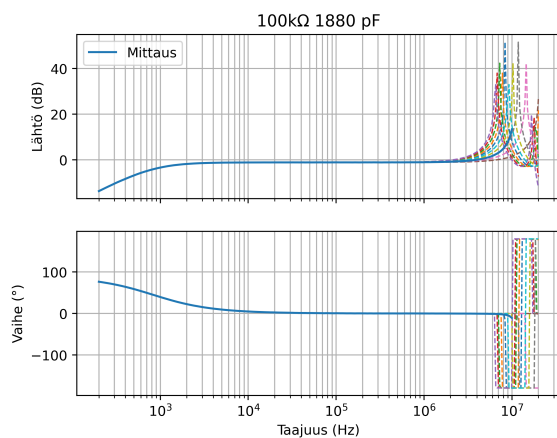
arvoilla mitatun taajuusvasteen molemmiin puolin. Induktanssin arvo pidettiin nimellisarvossa (Kuva 15). Sen jälkeen simuloituja taajuusvasteita verrattiin mitattuun taajuusvasteeseen python-skriptin avulla ensin matalammilla taajuuksilla, jossa vain kapasitanssin arvon muutoksella on vaikutusta. Mittattua taajuusvastetta lähimpänä olevan simulaation kapasitanssin arvoksi saatiin $C = 236\text{pF}$. (Kuva 16)

Kun kapasitanssi tunnettiin pidettiin sen arvo vakiona ja haarukoitiin vastaavasti induktanssi korkeammilla taajuuksilla (kuvat 17 ja 18). Induktanssin arvoksi saatiin $L = 0.317\mu\text{H}$. Koaksiaalikaapelin vastuksen arvon muuttaminen vaikutti vaihekäyrän muotoon resonanssin lähistöllä. Haarukoimalla eri vastuksen arvoilla simuloituja vaihekäyriä, koaksiaalikaapelin vastuksen arvoksi saatiin $R = 3\Omega$ (kuvat 19 ja 20).

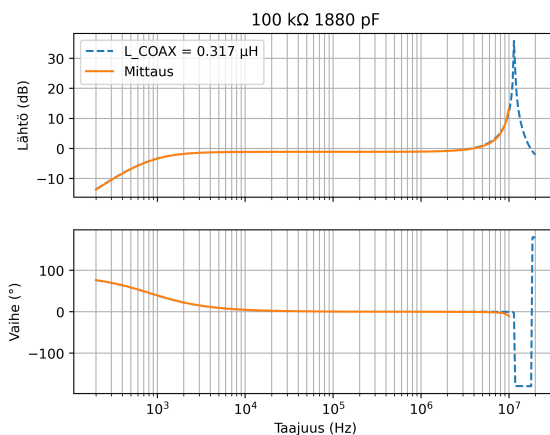
Simuloitu taajuusvaste saatiin vastaamaan melko hyvin mitattua taajuusvastetta haarukoiduilla koaksiaalikaapelin mallin arvoilla (Taulukko 2). Ongelmaksi tulosten vertailussa muodostuu kuitenkin oskilloskoopin taajuusalueen yläraja (10 MHz), joka on juuri mallinnetussa vasteessa havaittavan resonanssiipiikin alapuolella.

Taulukko 2. Haarukoidut parametrit koaksiaalikaapeleille

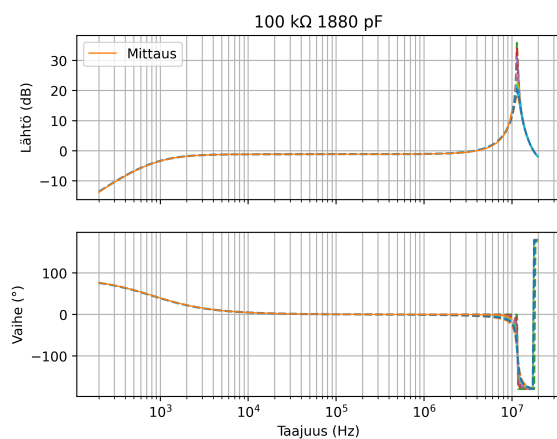
$$\begin{aligned} C &= 236\text{pF} \\ L &= 0.317\mu\text{H} \\ R &= 3\Omega \end{aligned}$$



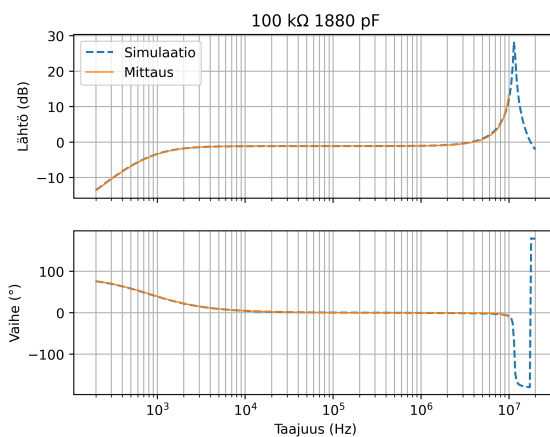
Kuva 17. Piirin simuloidut taajuusvasteet eri induktanssin arvoilla, vakio-kapasitanssilla $C = 236$ pF. Induktanssin kasvaessa resonanssiipiikki siirtyy matalammille taajuuksille.



Kuva 18. Python-skriptillä haarukoitu simuloitu taajuusvaste, jonka vahvistus on lähimpänä mitattua kaikilla taajuuksilla $f < 10$ MHz.



Kuva 19. Piirin simuloidut taajuusvasteet eri vastuksen arvoilla, vakio-kapasitanssilla $C = 236$ pF ja vakioinduktanssilla $L = 0.317$ μH.



Kuva 20. Python-skriptillä haarukoitu simuloitu taajuusvaste (katkovii-va), joka on lähimpänä mitattua taajuusvastetta kaikilla taajuuksilla $f < 10$ MHz.

3.4 Piirin aikakäytöksen mallintaminen

Mittausta vastaavan piirin vahvistus ja vaihe on saatu simuloitua, joten myös piirin aikakäyttö tunnetaan, sillä lineaariselle aikariippumattomalle systeemille aikakäyttö saadaan taajuusvasteen käänteisenä Fourier-muunnoksena. Simulaatioiden avulla voidaan siis selvittää millaisia herätevirtapulsseja mitatut vastejännitepulssit vastaavat. Piiriä simuloitiin LTspicellä ajan funktiona syöttämällä piiriin virtapulsseja, käyttäen aiemmassa kappaleessa määritettyjä komponenttien arvoja. Aiemmista mittauksista tiedetään, että herätevirtapulssit ovat muodoltaan nopeita tuplaeksponenttifunktioita eli muotoa

$$I(t) = I_o(e^{-at} - e^{-bt}) , \quad (3)$$

missä a ja b ovat pulssin aikavakioiden käänteislukuja. a vastaa pulssin nousureunaa, ja b pulssin laskureunaa. Virran määritelmästä

$$I(t) = \frac{dQ}{dt} , \quad (4)$$

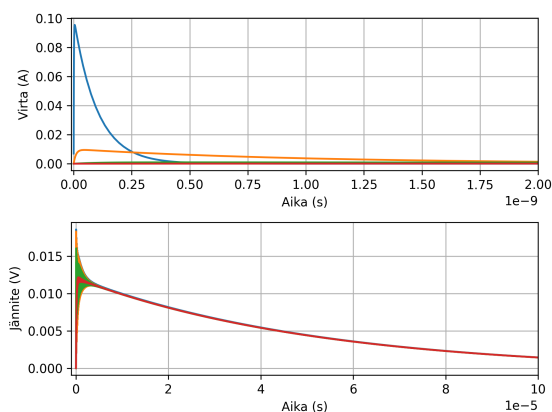
saadaan pulssin varaukselle

$$Q = \int_0^\infty I_o(e^{-at} - e^{-bt}) dt = I_o \left(\frac{1}{a} - \frac{1}{b} \right) . \quad (5)$$

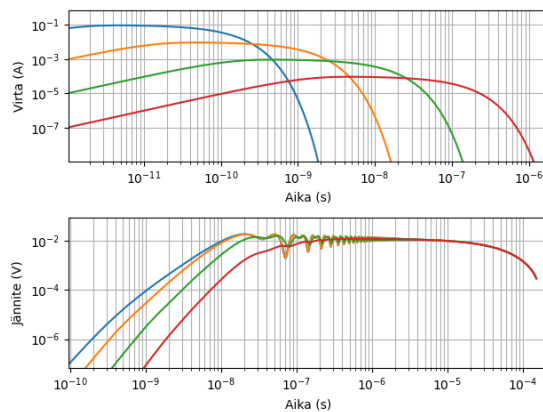
Jännitepulssit ovat samanmuotoisia, mutta paljon hitaampia. Olkoon jännitepulssien aikavakioiden käänteisluvut c ja d . Jännitepulssin aika-integraaliksi saadaan siten myös

$$\int_0^\infty V(t) dt = V_o \left(\frac{1}{c} - \frac{1}{d} \right) . \quad (6)$$

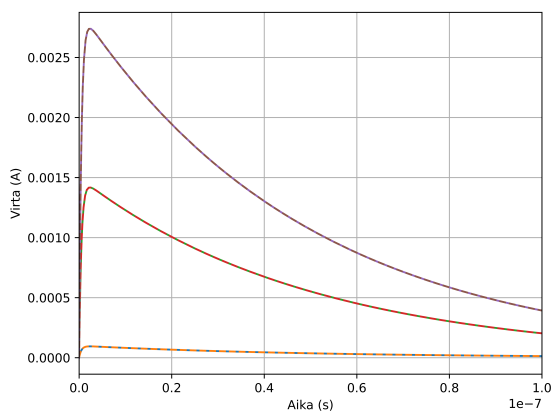
Aluksi huomattiin, että nopea virtapulssi korkealla amplitudilla voi tuottaa lähes samanlaisen jännitevasteen kuin hitaampi pulssi pienemmällä virralla, kunhan virtapulssin aikaintegraali eli varaus on sama (kuvat 21 ja 22). Herätevirtapulssin nopeus ei siis vaikuta vastejännitepulssin yleiseen muotoon alun oskillaatioita lukuunottamatta. Pienemmällä virtapulssin amplitudeilla, ja siten hitaammilla pulsseilla alun oskillaatio pienenee, mutta jos herätepulssi on liian hidaskä ($\frac{1}{b} > 1\mu s$), vastepulssin nousuaika kasvaa merkittävästi. Koska mitatuissa pulsseissa ei havaita oskillaatioita ja silti pulssien nousureuna on nopea, pulssin aikavakiot ovat simulaatioiden



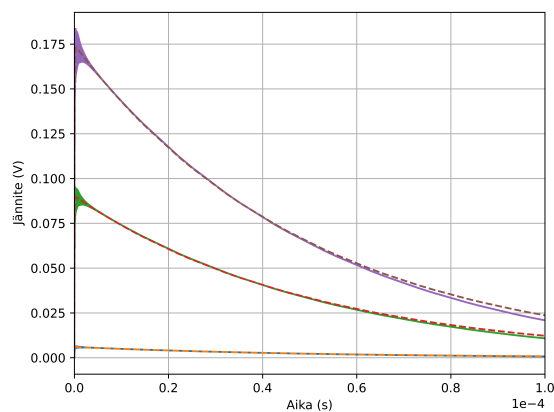
Kuva 21. Simuloidut herätevirtapuls-
sit vakiovarauksella ja pulsseja vastaa-
vaat vastejännitteet



Kuva 22. Simuloidut herätevirtapuls-
sit vakiovarauksella ja pulsseja vastaa-
vaat vastejännitteet logaritmisella astei-
kolla



Kuva 23. Simuloitujen virtapuls-
sien eksponenttisovitus



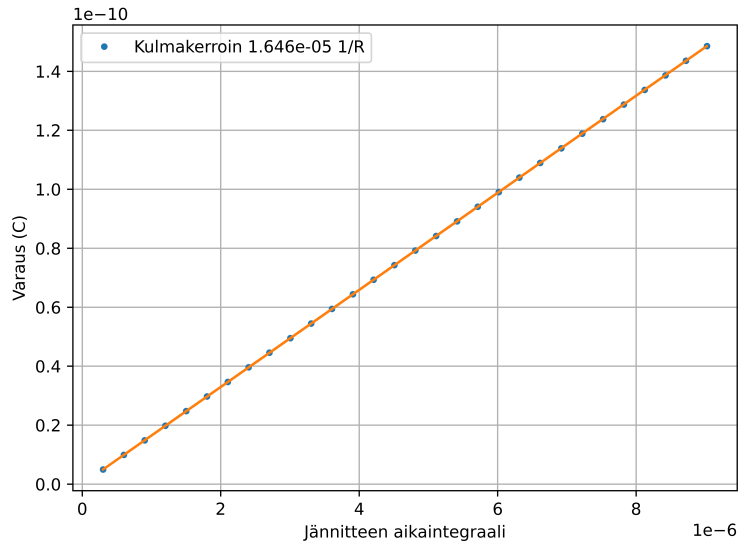
Kuva 24. Simuloitujen jännitepuls-
sien eksponenttisovitus

perusteella suuruusluokaltaan $50\text{ns} < \frac{1}{b} < 500\text{ns}$, $0.5\text{ns} < \frac{1}{a} < 5\text{ns}$.

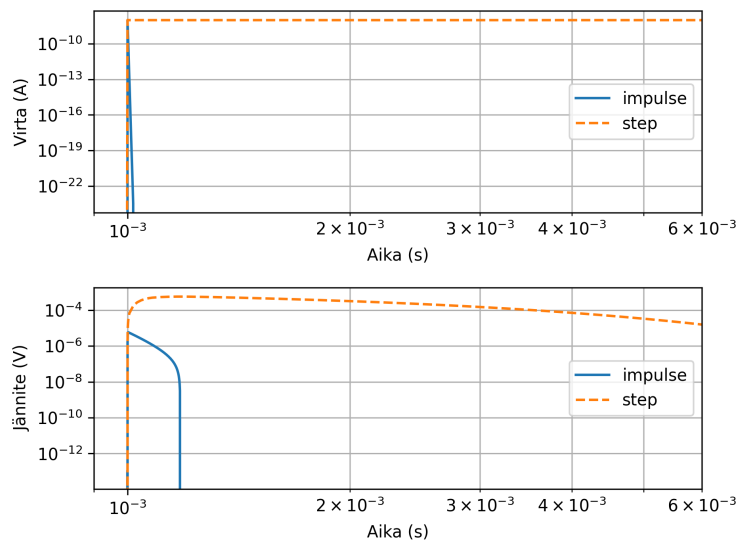
Kun simuloituihin herätevirta- ja vastejännitepulsseihin sovitetaan tuplaeksponenttifunktiot, saadaan pulssien integraaleista lineaarinen yhteys herätevirtapuls-
sin varauksen ja jännitepuls-
sin aikaintegraalin välille. (Kuvat 23, 24, 25)

Myös piirin askelvastetta tarkasteltiin simuloimalla (kuva 26). Huomataan, että askelherätteellä vastejännitepulssi on huomattavasti hitaampi kuin nopealla pulssiherätteellä.

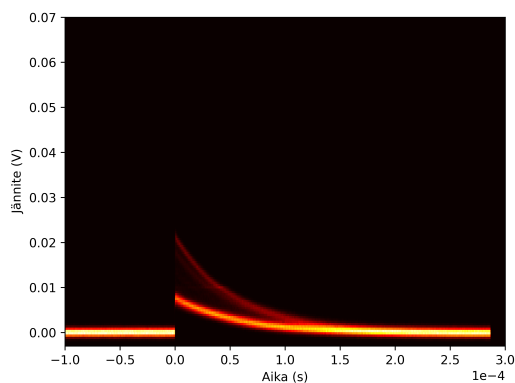
Nyt kun piirin herätteen ja vasteen välinen yhteys tunnetaan, voidaan mitatuista vastejännitepulsseista selvittää hiukkasosumasta kerätyn varauksen määrä.



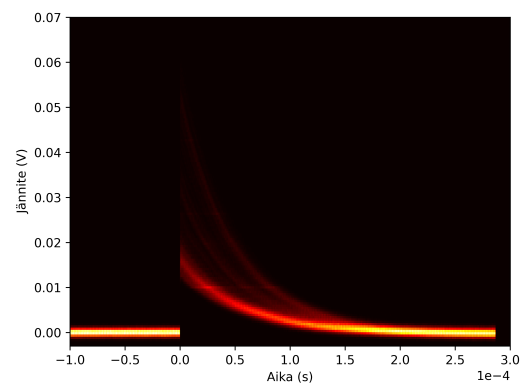
Kuva 25. Simuloitujen herätevirtapulssien varauksen ja jännitteen aikaintegraalin yhteys. Siniset pisteet vastaavat eksponenttifunktioiden integraaleja ja oranssi viiva on suoransovitus kulmakertoimella $k = 1.646 \times 10^{-5} \text{ 1}/\Omega$



Kuva 26. Piirin impulssi- ja askelvaste



Kuva 27. MOSFET 150 V



Kuva 28. MOSFET 300 V

4 Tulosten tarkastelu

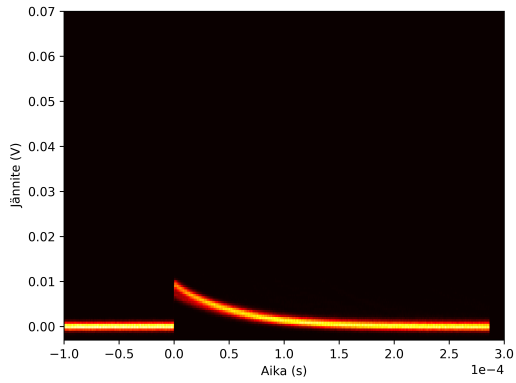
Tässä kappaleessa esitellään ensin muiden tekemien säteilynkestomittauksien tuloksia. Sen jälkeen piiristä simuloidun mallin avulla selvitetään mitattuja jännitepulsseja vastaavien herätevirtapulssien varaukset ja siten mittauksen virtapulssien varausjakauma. Lisäksi tarkastellaan vuotovirran portaittaisten nousujen lähistöllä syntyneiden pulssien varauksia.

4.1 Säteilynkestomittauksen visualisointi

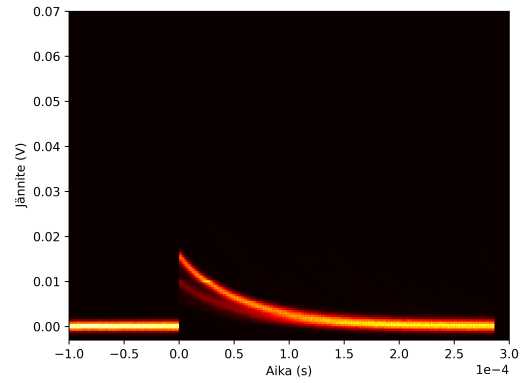
Suuremmilla estosuuntaisilla etujännitteillä hiukkasosumista syntyy enemmän korkeamman amplitudin pulsseja sekä SiC MOSFETeilla, että SiC-diodeilla. Mitatut jännitepulssit on esitetty kuvissa 27-30.

Suuremmilla etujännitteillä MOSFET:n vuotovirta alkaa kasvaa säteilytyksen vaikutuksesta. Vuotovirrassa tapahtuu nopeita portaittaisia nousuja. (Kuva 31)

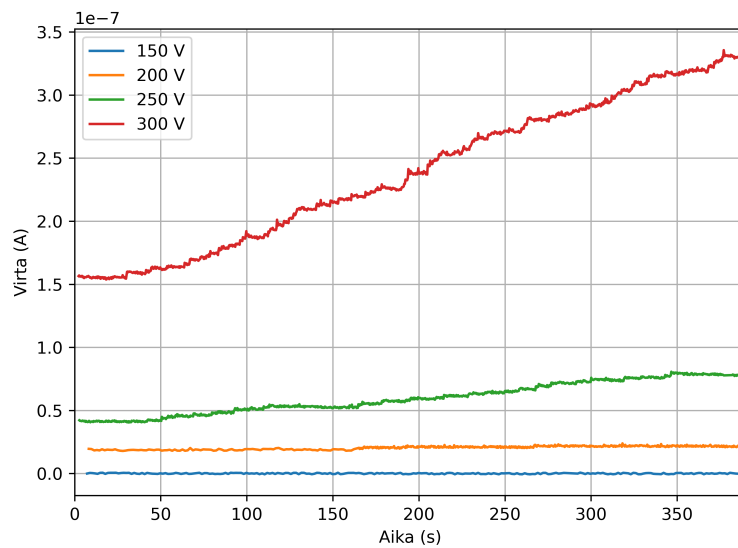
Diodin vuotovirta alkaa kasvaa tasaisesti kaikilla etujännitteillä 150 V - 300 V säteilytyksen alkaessa. (Kuvat 32 ja 33)



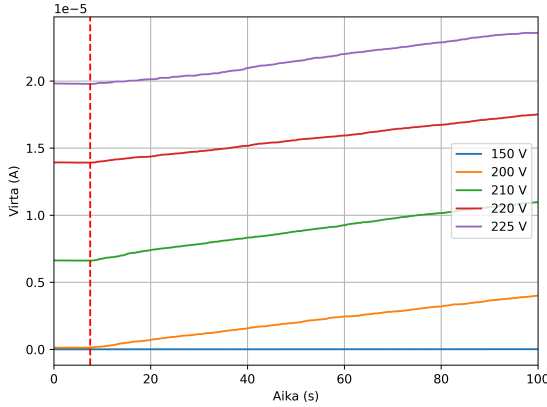
Kuva 29. Diodi 100 V



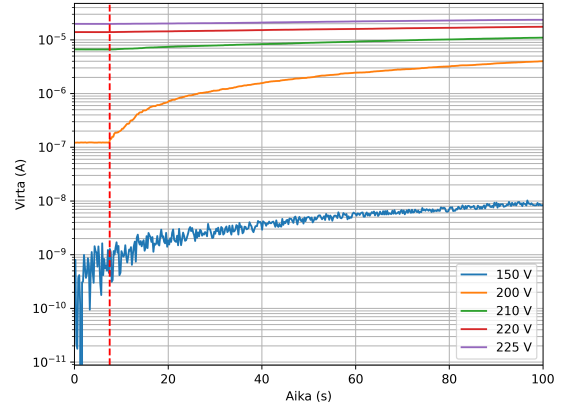
Kuva 30. Diodi 300 V



Kuva 31. MOSFET:n vuotovirta eri etujännitteillä



Kuva 32. Diodin vuotovirta alkaa kasvaa säteilytyksen alkaessa (punainen katkoviiva)



Kuva 33. Diodin vuotovirta logaritmisella asteikolla. Virta kasvaa hieman myös 150V etujännittellä

4.2 Säteilynkestomittausten analysointi mallin avulla

Mitatut jännitepulssit sovitettiin vastaavaan tuplaeksponenttifunktioon kuin simuloitujen pulssien (Kuva 34). Huomataan, että tuplaeksponenttifunktion aikavakio ei juurikaan muutu pulssin amplitudin kasvaessa, eli riippumatta amplitudista, jännitepulssien muoto ajan funktiona on sama. (Kuvat 35 ja 36)

Jännitepulssien eksponenttisovitusintegraaleista saadaan simuloidun mallin avulla selvitettyä herätevirtapulssien varaus. 250 V:n estosuuntaisessa etujännitteessä säteilytetyn MOSFET:n jännitepulssien integraalien ja herätevirtapulssien varausten jakaumat on esitetty kuvassa 37.

Mittauksessa käytetyn Au:n energiansiirto piikarbidissa on $LET_\rho = 94 \text{ MeVcm}^2/\text{mg}$, josta saadaan kertomalla 4H-SiC:n tiheydellä $\rho = 3.21 \cdot 10^3 \text{ mg/cm}^3$

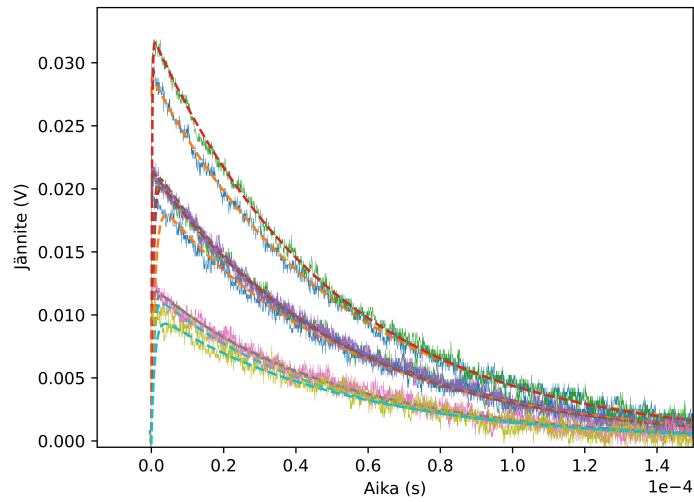
$$LET_l = LET_\rho \cdot \rho = 30.174 \text{ MeV}/\mu\text{m}. \quad (7)$$

Energiansiirrosta LET_l ja elektroni-aukkoparin luomisenergiasta ϵ saadaan yhden hiukkasosuman luoma elektroni-aukkoparien määrä n pituusyksikköä kohden

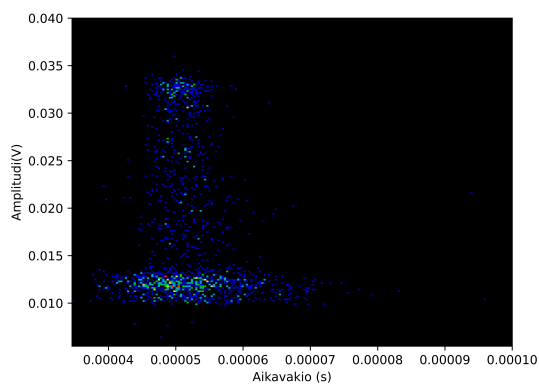
$$\frac{n}{l} = LET_\rho \cdot \epsilon \quad (8)$$

Kun parien määrä kerrotaan alkeisvarauksella q ,

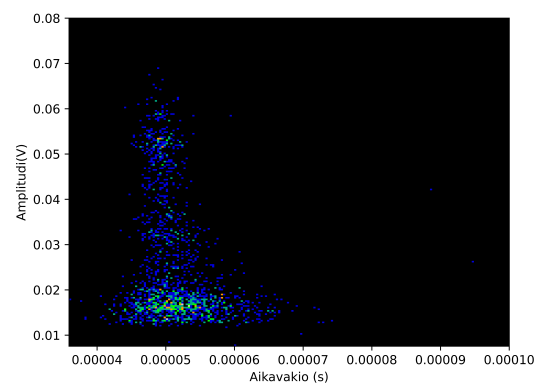
$$\frac{n}{l} \cdot q = \frac{Q}{l} = 0,664 \text{ pC}/\mu\text{m} \quad (9)$$



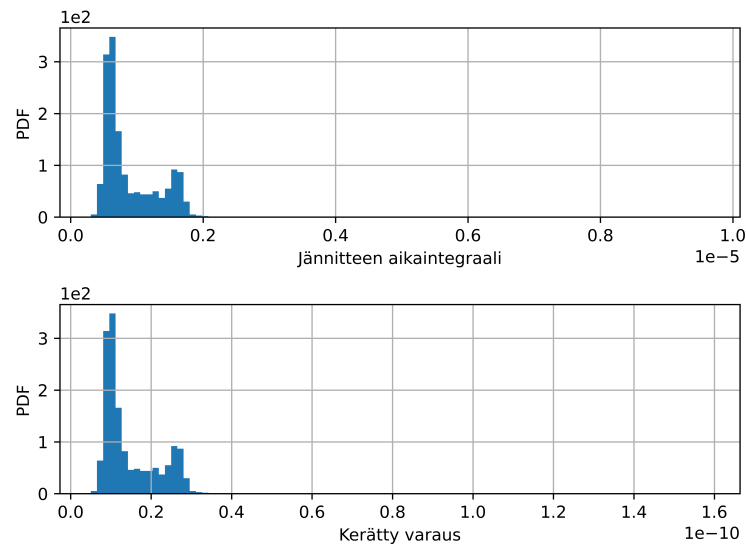
Kuva 34. Mitattujen jännitepulssien sovitus tuplaeksponenttifunktion $V_{ds} = 250V$



Kuva 35. Amplitudin vaikutus aikavakioon MOSFET $V_{ds} = 250V$



Kuva 36. Amplitudin vaikutus aikavakioon MOSFET $V_{ds} = 300V$

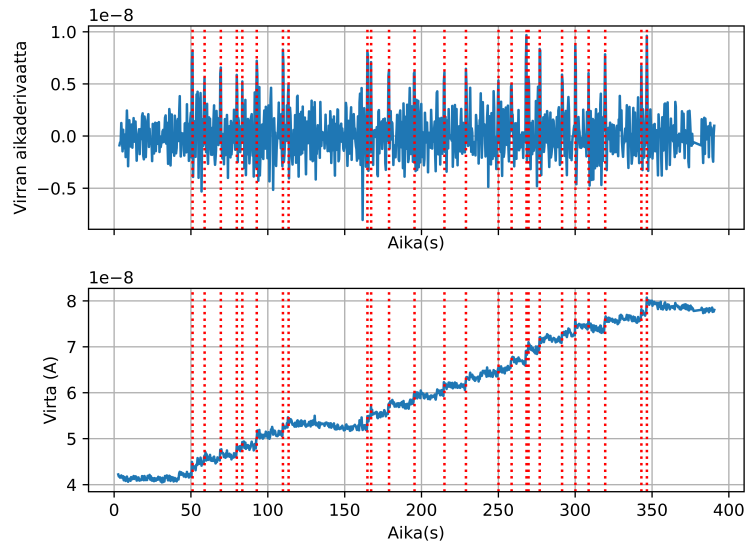


Kuva 37. Varausjakaumat MOSFET $V_{ds} = 250V$

saadaan luodun varauksen määrä pituusyksikköä kohden. Mitatun varausjakauman moodi on 10pC. Josta saadaan pituudeksi

$$l = \frac{10\text{pC}}{0,664\text{pC}/\mu\text{m}} = 15\mu\text{m}, \quad (10)$$

eli varausta kerätään n. 15 μm paksuudelta.

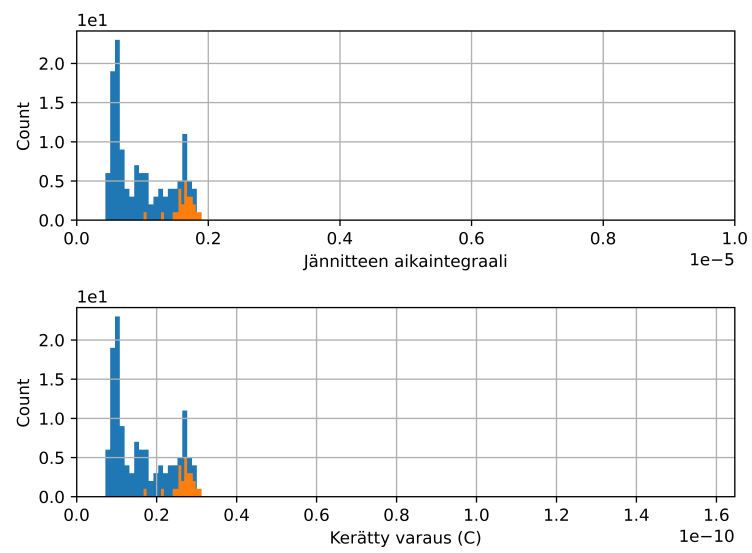


Kuva 38. Derivaatat

4.3 Vuotovirran kasvu ja suuremman varauksen pulssit

Vuotovirran portaittaisten nousujen sijainnit ajan funktiona selvitettiin Python-skriptin avulla derivoimalla vuotovirta ja etsimällä derivaatan huiput. Virtamittarin resoluutiosta johtuvat piikit suodatettiin pois liukuvan keskiarvoistuksen avulla. (Kuva 38)

Jokaiselle jännitepulssille tunnettiin myös samalla tietokoneella mitattu aikaleima, joten voitiin tarkastella vuotovirran nousujen lähistöllä syntyneitä pulsseja erikseen. Aikamittausten synkronointi ei ollut kuitenkaan täydellinen, joten portaiden aikaleimojen ympäristöstä $t = \pm 0.5s$ etsittiin suurimmat pulssit. Kaikkien pulssien ja virtaportaiden lähistöllä syntyneiden pulssien varausjakaumat on esitetty kuvassa 39. Jakaumia tarkastelemalla näyttäisi siltä, että pysyvät nousut vuotovirrassa johtuvat todennäköisesti pulsseista joilla on suurempi varaus.



Kuva 39. Kaikkien jännitepulssien ja virtapulssien varausten jakaumat sinisellä. Vuotovirran nousujen lähistöllä tapahtuneiden pulssien maksimit oranssilla

5 Johtopäätökset

Suodatinpiirin käytös saatiin mallinnettua mittaamalla piirin taajuusvaste ja haaru-koimalla arvot parasiittisille impedansseille SPICE-simulaatioiden ja Pythonin avulla. Piirin mallinnusta rajoitti funktiogeneraattorin taajuuden yläraja, jonka takia piirin resonanssiipiikki jäi mittausalueen ulkopuolelle. Piirin mallin avulla saatiin yhteys mitattujen vastejännitepulssien ja herätevirtapulssien varauksen välille. Tarkkaa herätepulssin muotoa ei kuitenkaan voitu selvittää, sillä piirin läpi mitattu jännitepulssi voi syntyä hyvin erilaisista herätteistä, kunhan herätepulssin varaus pysyy vakiona. Lisäksi herätevirtapulssit ovat todella nopeita, vain nanosekunnin suuruusluokkaa, joten piirin taajuusvaste olisi hyvä tuntea tarkemmin myös korkeammilla taajuuksilla.

Tarkastelemalla vuotovirran portaittaisten nousujen lähistöllä syntyneiden pulssien varauksia, vaikuttaisi siltä, että vuotovirran kasvun syynä ovat todennäköisesti pulssit joiden varaus on suurempi. Jotta tähän saisi varmuutta, tulisi pulssien ja vuotovirran aikaleimojen mittauksen olla tarkemmin synkronoitua.

Piirin avulla pystytään siis tarkastelemaan ioniosumista syntyneiden pulssien varauksien suuruutta, mutta tarkempaa tietoa syntyneistä pulsseista ei tällä mittausasetelmalla saada. Lisäksi piirin analysointiin pitäisi käyttää mittausasetelmaa, jolla saadaan tietoa myös piirin käytöksestä korkeammilla taajuuksilla.

Lähteet

- [1] U. of Concerned Scientists. *UCS Satellite Database 1-1-2022*. Viitattu: 22.4.2022. 2022. URL: http://www.ucsusa.org/satellite_database.
- [2] C. Leonard. *Challenges for Electronic Circuits in Space Applications*. 2017.
- [3] M. Xapsos. ”A Brief History of Space Climatology: From the Big Bang to the Present”. *IEEE Transactions on Nuclear Science* 66.1 (2019), s. 17–37. DOI: 10.1109/TNS.2018.2885649.
- [4] W. Li ja M. Hudson. ”Earth’s Van Allen Radiation Belts: From Discovery to the Van Allen Probes Era”. *Journal of Geophysical Research: Space Physics* 124.11 (), s. 8319–8351. DOI: <https://doi.org/10.1029/2018JA025940>. URL: <https://agupubs.onlinelibrary.wiley.com/doi/abs/10.1029/2018JA025940>.
- [5] T. Kimoto ja J. A. Cooper. *Fundamentals of silicon carbide technology: growth, characterization, devices and applications*. Chapter 2. John Wiley & Sons, 2014.
- [6] M. Östling, R. Ghandi ja C.-M. Zetterling. ”SiC power devices — Present status, applications and future perspective”. Teoksessa: *2011 IEEE 23rd International Symposium on Power Semiconductor Devices and ICs*. 2011, s. 10–15. DOI: 10.1109/ISPSD.2011.5890778.
- [7] A. Mashlay. *The Creation and Potential Cell Structures of SiC Devices*. viitattu 3.5.2022. 2017.
- [8] A. Elasser ja T. Chow. ”Silicon carbide benefits and advantages for power electronics circuits and systems”. *Proceedings of the IEEE* 90.6 (2002), s. 969–986. DOI: 10.1109/JPROC.2002.1021562.
- [9] X. She ym. ”Review of Silicon Carbide Power Devices and Their Applications”. *IEEE Transactions on Industrial Electronics* 64.10 (2017), s. 8193–8205. DOI: 10.1109/TIE.2017.2652401.
- [10] G. F. Knoll. *Radiation detection and measurement*. John Wiley & Sons, 2010.
- [11] P. Sigmund. *Stopping of heavy ions: a theoretical approach*. Vol. 204. Springer Science & Business Media, 2004.

- [12] M. Xapsos. "Applicability of LET to single events in microelectronic structures". *IEEE Transactions on Nuclear Science* 39.6 (1992), s. 1613–1621. DOI: 10.1109/23.211343.
- [13] S. Chaudhuri, K. Zavalla ja K. Mandal. "Experimental determination of electron-hole pair creation energy in 4H-SiC epitaxial layer: An absolute calibration approach". *Applied Physics Letters* 102 (tammikuu 2013), s. 31109.
- [14] A. F. Witulski ym. "Single-Event Burnout Mechanisms in SiC Power MOSFETs". *IEEE Transactions on Nuclear Science* 65.8 (2018), s. 1951–1955. DOI: 10.1109/TNS.2018.2849405.
- [15] D. R. Ball ym. "Ion-Induced Energy Pulse Mechanism for Single-Event Burnout in High-Voltage SiC Power MOSFETs and Junction Barrier Schottky Diodes". *IEEE Transactions on Nuclear Science* 67.1 (2020), s. 22–28. DOI: 10.1109/TNS.2019.2955922.
- [16] k. Martinella Corinna. "Single-event radiation effects in silicon carbide power MOSFETs". Artikkeliväitöskirjan yhteenveto-osa ja 3 eripainosta. Väitöskirja. Jyväskylä: Väitöskirja : 2021. URL: <http://urn.fi/URN:ISBN:978-951-39-8726-8>.
- [17] C. Martinella ym. "Heavy-Ion Microbeam Studies of Single-Event Leakage Current Mechanism in SiC VD-MOSFETs". *IEEE Transactions on Nuclear Science* 67.7 (2020), s. 1381–1389. DOI: 10.1109/TNS.2020.3002729.

A Parhaan simulaation haarukointi

```
import os
import csv
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt

#Haarukoi hakemistosta olevista bodesimulaatioista sen joka on lähimpänä
#hakemistossa olevaa bodemittausta ja
#plottaa parhaan simulaation ja mittauksen

#Hakemisto josta tiedostot haetaan
PATH = '...\simulaatiot\940p100khaarukointi3\'

#Simulaatiot nimettävä ***sim.csv ja mittaukset ***mit.csv
fileNames = os.listdir(PATH)
fileNames2 = [file for file in fileNames if 'sim.csv' in file]
fileNames3 = [file for file in fileNames if 'mit.csv' in file]

#taajudet joissa mittauksia ja simulaatioita vertaillaan
taajudet=np.interp([6e+6, 6.2e+6, 6.4e+6, 6.6e+6, 6.8e+6, 7e+6, 8e+6,
9e+6, 9.5e+6, 9.6e+6, 9.7e+6, 9.8e+6, 9.9e+6, 10e+6])

#matalien taajuuksien haarukointi
#taajuudet=np.interp([1e+3, 5e+3, 1e+4, 5e+4, 1e+5, 5e+5])

#lukee mitatun csv:n
for file in fileNames3:
```

```
df = pd.read_csv(PATH + file,header=None,skiprows=21)

#lukee simulaatioiden csv:n
for file in fileNames2:

    #Laskee ajojen määrän
    string_to_search1="Run"
    stepcount=0

    with open(PATH + file, 'r') as read_obj:
        # Read all lines in the file one by one
        for line in read_obj:
            # For each line, check if line contains the string
            if string_to_search1 in line:
                stepcount += 1

    #Laskee yksittäisen ajon pituuden
    string_to_search = "Run: 2"
    line_number=0

    with open(PATH + file, 'r') as read_obj:
        # Read all lines in the file one by one
        for line in read_obj:
            # For each line, check if line contains the string
            line_number += 1
            if string_to_search in line:
                break

    line_number=line_number-2
    ajonpituus = line_number-1
    dfs = list(range(0, stepcount))
```

```

#Tekee ajoista dataframen
for i in range (0,stepcount):
    riviskip=2+(line_number)*i
    dfs[i] = pd.read_csv(PATH + file, sep='[\t,]',
    header=None,skiprows=riviskip,
    nrows=ajonpituus, engine='python')
    dfs[i] = dfs[i].replace({'dB':'', '\u00b0':'', '\(':'', '\)':''}, regex=True)
    dfs[i] = dfs[i].astype(float) #muuttaa datatyypin takaisin numeroiksi

#Haarukoi lähimpänä olevan simulaation
dfmit = df
pienin=1000000000

# etsii simulaation joka on lähimpänä mittausta
for i in range (0,stepcount):
    dfsim = dfs[i]
    erotus=0

    #taajuuksien haarukointi
    ymit= taajuudet, dfmit[0], dfmit[2])
    ysim= taajuudet, dfsim[0], dfsim[1])

    erotus = ((ymit-ysim)/(ysim))*2 #erotus vektorin alkioittain
    erotus2 = np.sum(erotus)          #summaa vektorin alkioit
    if erotus2 < pienin:
        pienin = erotus2
        parhaanindeksi = i #tallentaa mittausta lähimpänä olevan simulaation
        #indeksin

```

```
#plottaus
#rivien lukumäärä ennen parasta simulaatiota
riviskip=2+(line_number)*parhaanindeksi
dfs[parhaanindeksi] = pd.read_csv(PATH + file,
                                sep='[\t,]',
                                header=None,
                                skiprows=riviskip,
                                nrows=ajonpituus,
                                engine='python')
dfs[parhaanindeksi] = dfs[parhaanindeksi].replace(
    {'dB':'', '\u00b0':'', '\(:'', '\)'':'', },
    regex=True)
dfs[parhaanindeksi] = dfs[parhaanindeksi].astype(float)
dfsim = dfs[parhaanindeksi]
ax1.semilogx(dfsim[0],dfsim[1], '--', label=file.split('.')[0])
ax2.semilogx(dfsim[0],dfsim[2], '--', label=file.split('.')[0])
```

B Simulaatioiden eksponenttisovitus

```
import os
import csv
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from lmfit import Model

#Sovittaa tuplaeksponenttifunktiot simuloituihin jännite- ja virtapulsseihin
#ja sovittaa suoran virta- ja jännitepulssien integraalien välille.
#Plottaa myös sovitukset ja suoran

def exponential(x, a, tfall, trise):
    return a*(np.exp(-(x/tfall)) - np.exp(-(x/trise)))

def integral(a, tfall, trise): #tuplaeksponenttifunktion integraali
    return a*(tfall-trise)

def suora(x, k):
    return k*x

f, (ax3) = plt.subplots(nrows=1,sharex=False)

PATH = 'C:\\...\\virtavsjannite2\\'
fileNames = os.listdir(PATH)
fileNames3 = [file for file in fileNames if 'Run' in file]
```

```

virtaintegraalit=np.zeros(0)
janniteintegraalit=np.zeros(0)

for file in fileNames3:
    df = pd.read_csv(PATH + file,header=None,skiprows=1,
                    sep=",", engine='python')
    x=df[0]
    y=df[2]
    y2=df[1]

    gmodel = Model(exponential)
    result = gmodel.fit(y, x=x, a=0.001, tfall=50*10**-9, trise=0.5*10**-9)
    #ax1.plot(x,y)
    #ax1.plot(x, result.best_fit,'--')
    iparametrit=result.best_values

    gmodel = Model(exponential)
    result2 = gmodel.fit(y2, x=x, a=0.1, tfall=50*10**-6, trise=0.5*10**-6,
                        nan_policy='propagate')
    #ax2.plot(x,y2)
    #ax2.plot(x, result2.best_fit,'--')
    vparametrit=result2.best_values

    janniteintegraali=integral(vparametrit.get('a'),vparametrit.get('tfall'),
                              vparametrit.get('trise'))
    janniteintegraalit=np.append(janniteintegraalit,janniteintegraali)

    virtaintegraali=integral(iparametrit.get('a'),iparametrit.get('tfall'),
                              iparametrit.get('trise'))
    virtaintegraalit=np.append(virtaintegraalit,virtaintegraali)

    siirtofunktio=janniteintegraali/virtaintegraali

```

```
gmodel = Model(suora)
result3 = gmodel.fit(virtaintegraalit, x=janniteintegraalit, k=1)

ax3.plot(janniteintegraalit, virtaintegraalit, '.')
ax3.plot(janniteintegraalit, result3.best_fit, '-')
```

```
siirtofunktio2=result3.best_values
kulmakerroin=siirtofunktio2.get('k')
kulmakerroinstring=str('{:0.3e}'.format(kulmakerroin))
```

```
#ax1.set_xlabel('Aika (s)')
#ax1.set_ylabel('Virta (A)')
#ax1.set_xlim([0, 0.0000001])
#ax1.grid(True,which="both",ls="-")
#ax1.ticklabel_format(axis="x", style="sci", scilimits=(0,0))
```

```
#ax2.set_xlabel('Aika (s)')
#ax2.set_ylabel('Jännite (V)')
#ax2.set_xlim([0, 0.00015])
#ax2.grid(True,which="both",ls="-")
#ax2.ticklabel_format(axis="x", style="sci", scilimits=(0,0))
```

```
ax3.set_xlabel('Jännitteen aikaintegraali')
ax3.set_ylabel('Varaus (C)')
ax3.grid(True,which="both",ls="-")
ax3.legend(["Kulmakerroin " + kulmakerroinstring + " 1/R"])
ax3.ticklabel_format(axis="both", style="sci", scilimits=(0,0))
```

```
#lgd = ax1.legend(loc='center right', bbox_to_anchor=(1.35, 0.3))
plt.tight_layout()
plt.show()
```

```
f.savefig(PATH + 'simeksponenttifitsuora.png', dpi=1000, format='png')
```


C Luokat datan muotoiluun

```
import re
import matplotlib.pyplot as plt
import pandas as pd
class Virtadata:

    def getStartTime(file):

        with open(file) as f:
            first_line = f.readline()

            first_line = first_line.replace('Thu Mar 14 ', '')
            first_line = first_line.replace(' 2019', '')
            first_line=sum(x * int(t) for x, t in zip([3600, 60, 1],
            first_line.split(":")))

            time=first_line
            return time

    def getVirtaData(file):

        df = pd.read_csv(PATH + file,header=None,skiprows=6, sep="\t",
        engine='python')
        time=df[0]
        current=df[2]

        return time, current

    def plotVirtadata(file):
```

```

time=Virtadata.getStartTime(file)
f, (ax1) = plt.subplots(nrows=1,sharex=True)
df = pd.read_csv(file,header=None,skiprows=6, sep="\t", engine='python')
ax1.plot(df[0]+time,df[2],'-')

import re
import matplotlib.pyplot as plt
import pandas as pd
class Jannitedata:

    def getStartTime(file):

        f=open(file)
        lines=f.readlines()
        line=lines[3]
        line2=lines[6]

        line = line.replace('#Date Time: 2019-03-14 ', '')
        line=line.split('::')

        line= line[0] + ":" + line[1]
        line.replace('\n', '')
        time=sum(x * float(t) for x, t in zip([3600, 60, 1], line.split(":")))
        time2=line2.split(",")
        time2=float(time2[0])
        time=time+time2

    return time

    def getJanniteData(file):

        df = pd.read_csv(PATH + file,header=None,skiprows=7, sep="," , engine='python')
        time=df[0]
        voltage=df[1]

```

```
return time, current

def plotJannitedata(file):
    f, (ax1) = plt.subplots(nrows=1,sharex=True)
    df = pd.read_csv(file,header=None,skiprows=7, sep=",", engine='python')
    ax1.plot(df[0],df[1],'-')
```


D Mittausten sovitus ja kerääminen samaan tiedostoon

```
import os
import csv
import pandas as pd
import numpy as np
from lmfit import Model, Parameter
from jannitedata import Jannitedata
from virtadata import Virtadata

def exponential(x, a, tfall, trise):
    return a*(np.exp(-(x/tfall)) - np.exp(-(x/trise)))

def integral(a, tfall, trise): #tuplaeksponenttifunktion integraali
    return a*(tfall-trise)

file2="...\mosfet\VU_MOSFET_uncoated_SN056_run012_irrad.txt"
virta aika=Virtadata.getStartTime(file2)

PATH = 'C:\\...\\data\\scope_run012\\'
fileNames = os.listdir(PATH)
fileNames3 = [file for file in fileNames if 'acq' in file]

virtaintegraalit=np.zeros(0)
janniteintegraalit=np.zeros(0)
```

```

amplitudit=np.zeros(0)
tfallit=np.zeros(0)
triset=np.zeros(0)
aikaleimat=np.zeros(0)
runnumbers=np.zeros(0)
runnumber=0

for file in fileNames3:

    runnumber=runnumber+1
    runnumbers=np.append(runnumbers,runnumber)
    print(runnumber)
    df = pd.read_csv(PATH + file,
                     header=None,skiprows=7, sep=",", engine='python')

    keskiarvo=df[1].iloc[[0,30]].mean(axis=0) #nolla-akselin kalibrointi
    df[1]=-df[1]+keskiarvo

    fitinaloitus=df[1].idxmax(1)-10
    x=df[0].iloc[fitinaloitus:]
    y=df[1].iloc[fitinaloitus:]

    gmodel = Model(exponential)
    result = gmodel.fit(y, x=x, a=Parameter('a',value=0.1,min=0),
                       tfall=Parameter('tfall',value=50*10**-6,min=0),
                       trise=Parameter('trise',value=0.5*10**-6,min=0),
                       nan_policy='propagate')

    vparametrit=result.best_values
    amplitudit=np.append(amplitudit,vparametrit.get('a'))
    tfallit=np.append(tfallit,vparametrit.get('tfall'))
    triset=np.append(triset,vparametrit.get('trise'))

    janniteintegraali=integral(vparametrit.get('a'),vparametrit.get('tfall'),

```

```
        vparametrit.get('trise'))
janniteintegraalit=np.append(janniteintegraalit,janniteintegraali)

aikaleima=Jannitedata.getStartTime(PATH+file)-virta aika
aikaleimat=np.append(aikaleimat,aikaleima)

runnumbers=runnumbers.astype(int)

datat= {'runnumbers': runnumbers,
        'janniteintegraalit': janniteintegraalit,
        'amplitudit': amplitudit,
        'tfallit': tfallit,
        'triset':triset,
        'aikaleimat':aikaleimat}

outputdf=pd.DataFrame(data=datat,
                      columns= ["runnumbers","aikaleimat","janniteintegraalit",
                               'amplitudit','tfallit','triset'])

outputdf.to_csv(PATH + 'run_dataframe.csv',sep="\t",index=False)
```


E Vuotovirran nousujen ja suurempien jännitepulssien aikojen vertailu

```
import pandas as pd
import os
import numpy as np
import matplotlib.pyplot as plt
from scipy import signal, misc
from jannitedata import Jannitedata
from virtadata import Virtadata

aikamax=0.5 #kuinka monta sekuntia piikin jälkeen otetaan huomioon
aikamin=0.5 #kuinka monta sek ennen piikkiä
peakheight=0.48*10**-8 # derivaattapiikin korkeus

PATH2='C:\\...\\data\\scope_run010\\'
file2="run_dataframe.csv"
PATH='C:\\...\\data\\virtadatat\\mosfet\\'
file='VU_MOSFET_uncoated_SN056_run010_irrad.txt'

df = pd.read_csv(PATH + file,header=None,skiprows=10, sep="\s", engine='python')

x=df[0]
y=df[2]

dx=x.diff()
dy=y.diff()

derivaatta=dy/dx
```

```

derivaattarolling=derivaatta.rolling(2).sum()
peaks=signal.find_peaks(derivaattarolling,height=peakheight)
virta_ajat=x[peaks[0]]

df2 = pd.read_csv(PATH2 + file2, header=None, skiprows=1, sep="\s", engine='python')

runnumber=0
janniteajat=np.zeros(0)
janniteintegraalit=np.zeros(0)
integrmaksimit=np.zeros(0)
maksimienajat=np.zeros(0)

for aika in virta_ajat:
    tempjanniteintegraalit=np.zeros(0)
    maksiminaika=0
    for i in range(len(df2)):
        janniteaika = df2[1].iloc[i]
        erotus = aika - janniteaika
        if erotus > -aikamax and erotus < aikamin :
            janniteintegraalit=np.append(janniteintegraalit,
                                         df2[2].iloc[i])
            tempjanniteintegraalit=np.append(tempjanniteintegraalit,
                                             df2[2].iloc[i])
            if df2[2].iloc[i] == np.max(tempjanniteintegraalit) :
                maksiminaika=df2[1].iloc[i]-aika
    if tempjanniteintegraalit.size != 0 :
        integrmaksimit=np.append(integrmaksimit,
                                 np.max(tempjanniteintegraalit))
    maksimienajat=np.append(maksimienajat,maksiminaika)

f, (ax1, ax2) = plt.subplots(nrows=2,sharex=False)

```

```
k=1.646*10**(-5) #piirin vakio
virtaintegraalit=janniteintegraalit*k
virtamaksimit=integrmaksimit*k
num_bins=20
ax1.set_xlim(left=0.0,right=1e-5)
ax2.set_xlim(left=0.0,right=1.646*10**(-5)*1e-5)

ax1.hist(janniteintegraalit, num_bins)
ax1.hist(integrmaksimit, num_bins)
ax2.hist(virtaintegraalit, num_bins)
ax2.hist(virtamaksimit, num_bins)

ax1.set_xlabel('Jännitteen aikaintegraali')
ax1.set_ylabel('Count')
ax1.grid(True,which="both",ls="-")
ax1.ticklabel_format(axis="both", style="sci", scilimits=(0,0))

ax2.set_xlabel('Kerätty varaus (C)')
ax2.set_ylabel('Count')
ax2.grid(True,which="both",ls="-")
ax2.ticklabel_format(axis="both", style="sci", scilimits=(0,0))

plt.tight_layout()
plt.show()
f.savefig(PATH2 + 'jakaumatpeaks.png', dpi=1000, format='png')
```


F Vuotovirran ja sen derivaatan kuvaajat

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from scipy import signal, misc

PATH2='C:\\...\\data\\scope_run012\\'
PATH='C:\\...\\data\\virtadatat\\mosfet\\'
file='VU_MOSFET_uncoated_SN056_run012_irrad.txt'
peakheight=0.48*10**-8
df = pd.read_csv(PATH + file,header=None,skiprows=10, sep="\s", engine='python')

x=df[0]
y=df[2]

dx=x.diff()
dy=y.diff()

derivaatta=dy/dx

derivaattarolling=derivaatta.rolling(2).sum()

peaks=signal.find_peaks(derivaattarolling,height=peakheight)
f, (ax1, ax2) = plt.subplots(nrows=2,sharex=True)
ax1.plot(x,derivaattarolling)
ax2.plot(x,y)

for peak in peaks[0]:
    print(peak)
    ax1.axvline(x=x[peak],linestyle=':',color='r')
```

```
ax2.axvline(x=x[peak],linestyle=':',color='r')

ax1.set_xlabel('Aika(s)')
ax1.set_ylabel('Virran aikaderivaatta')

ax2.set_xlabel('Aika(s)')
ax2.set_ylabel('Virta (A)')

ax1.grid(True,which="both",ls="-")
ax2.grid(True,which="both",ls="-")
plt.tight_layout()
plt.show()
f.savefig(PATH2 + 'derivaatat.png', dpi=1000, format='png')
```