



**Departement Industriële Wetenschappen en Technologie**  
**Masterproef 2010-2011**

# **Ontwerp van een digitale filter voor dispersiecorrectie**

*Kevin Peeters*

**Promotoren:** ir. Wim Beaumont, UA-FYS  
ing. Koen Lostrie, KdG-IWT

Proefschrift tot het behalen van de graad van  
Master of Science in de Industriële Wetenschappen  
Elektronica-ICT afstudeerrichting ICT  
Hoboken, juni 2011

## Voorwoord

Ongeveer één jaar geleden, stelde KdG-IWT hun master-eindwerken voor. Een van de onderwerpen die werden aangeboden, was het onderzoeken van een coaxiale kabel waarop dispersie optreedt. Deze dispersie zorgt voor een timingprobleem. De dispersie zal aan de hand van een filter moeten worden onderdrukt.

Dit onderwerp sprak me direct aan omwille van de verschillende wetenschappelijke domeinen die aanbod kwamen: DSP, FPGA, (digitale) filters. Ook het feit dat het onderwerp voortkwam vanuit het CERN, wekte mijn interesse. Ik wist ongeveer waarmee ze op CERN allemaal bezig waren, namelijk deeltjesfysica. Na wat zoekwerk bleek al snel dat CERN voor nog veel meer andere zaken verantwoordelijk is, zoals bijvoorbeeld het World Wide Web. Het onderzoek naar nieuwe elementaire deeltjes heeft naast fysica, ook elektronica en ICT nodig om resultaten te boeken. Elektronica en ICT zorgen tijdens de experimenten voor het uitlezen en het verwerken van data.

Voordat er verdergegaan wordt met het onderzoek van de coaxiale kabel en het ontwerp van een filter voor de signaalcorrectie, wil ik de kans grijpen om enkele mensen te bedanken. Als eerste wil ik mijn promotor ir. W. Beaumont bedanken. Hij was altijd bereid om extra uitleg te geven. Ik heb veel van hem geleerd op professioneel vlak. Daarnaast wil ik ook mijn schoolpromotor ing. K. Lostrie bedanken voor zijn deskundige begeleiding en zijn hulp bij problemen. Ik wil hem ook bedanken om mij de mogelijkheid te geven om aan dit veelzijdige onderwerp te werken. Vervolgens nog een woord van dank aan de mensen van de onderzoeksgroep van de elementaire fysica, UA, KdG en mijn klasgenoten. Ten slotte wil ik mijn ouders en familie bedanken voor hun steun tijdens het onderzoek.

Kevin Peeters  
Hoboken, mei 2011

## Opdrachtomschrijving

De CASTOR detector te CERN maakt gebruik van het Cherenkov effect om elementaire deeltjes te detecteren. De lichtpulsen worden met behulp van een fotomultiplier (PMT) omgezet in een elektrisch signaal. De PMT genereert pulsen die korter zijn dan 15ns. Vervolgens worden deze getransporteerd via een 12 m lange coaxiale kabel naar een uitleeskaart (QIE). Deze zal het signaal over een tijdsinterval van 25ns integreren. Door dispersie is de puls langer dan 25ns wanneer het bij de QIE-kaart aankomt. Dit geeft problemen bij de analyse van de data. Daarom moet de puls terug worden getransformeerd naar een puls van minder dan 25ns.

Om dit te bereiken kunnen er zowel in het analoge gedeelte (voorversterker) als in het digitale gedeelte (FPGA, signaal bewerking) maatregelen worden genomen.

De opdracht van de thesis is om de effecten van de dispersie zo goed mogelijk te compenseren. Daarbij zal gebruik gemaakt worden van:

- elektriciteitsleer
- FPGA programmering
- (digitale) signaalverwerking

## Samenvatting

In dit werk zal onderzocht worden in welke mate een transmissielijn verantwoordelijk is voor de ontstane signaalstaarten veroorzaakt door dispersie. Daarvoor is een uitvoerig onderzoek van de coaxiale transmissielijn nodig. Er is onderzoek gedaan naar hoe een transmissielijn gemodelleerd kan worden, hoe de transferfunctie opgesteld kan worden, welke verliezen aanwezig zijn op de lijn, fasesnelheid, dispersie en reflecties. Het doel van dit onderzoek is het gedrag van de coaxiale kabel zo goed mogelijk te kunnen voorspellen.

Vervolgens moeten de signaalstaarten onderdrukt worden. Hiervoor zal gebruik gemaakt worden van digitale filters, namelijk FIR filters. Er zal een specificatie opgesteld moeten worden zodat de filtercoëfficiënten bepaald kunnen worden. Daarna moet de filter geïmplementeerd worden op een FPGA in VHDL. Hierbij zal optimalisatie van het resourcegebruik nodig zijn.

## Inhoudstafel

Voorwoord .....	ii
Opdrachtschrijving .....	iii
Samenvatting.....	iv
Gebruikte afkortingen .....	viii
1. Inleiding.....	1
1.1 CERN .....	1
1.1.1 Large Hadron Collider of LHC .....	2
1.1.2 CMS of Compact Muon Solenoid .....	3
1.2 CASTOR .....	5
1.3 PMT.....	7
1.4 QIE.....	8
1.5 CASTOR read-out en triggerarchitectuur / data acquisitie hardware.....	10
1.6 De triggerlogica.....	12
2 Het probleem .....	15
2.1 De probleemstelling .....	15
2.2 De opdracht .....	15
2.3 Het plan van aanpak .....	16
3 Transmissielijn.....	17
3.1 Fasesnelheid en groepssnelheid.....	22
3.2 Dispersie .....	23
3.3 Reflecties en afsluiting van de lijn .....	24
4 Analyseren van de coaxiale transmissielijn .....	25
4.1 DRAKA – coaxiale kabel .....	25
4.2 Rise time – Bandbreedte .....	25
4.3 De frequentieafhankelijkheid van de demping .....	27
4.4 De frequentieafhankelijkheid van de fase.....	30
4.5 Vierpuntsmeting voor het bepalen van de weerstand van de kabel .....	31
5 Opstellen van het mathematische model van de kabel .....	32
5.1 Gebruikte software: Matlab .....	32
5.2 Basic fitting op gemeten amplitude- en fasekarakteristiek .....	32
5.3 Model aan de hand van een transferfunctie .....	36

5.3.1	Besluit .....	38
6	FIR filter .....	39
6.1	Wat is een FIR filter?.....	39
6.2	Alternatief: Feedback filter .....	41
6.3	Bepalen van de filtercoëfficiënten .....	42
6.4	De orde van de filter .....	43
6.4.1	Besluit .....	45
6.5	Methodiek voor het ontwerp van de filters .....	46
7	Software .....	48
7.1	Dataverwerking met C++ .....	48
7.2	Software implementatie van een FIR filter .....	49
7.3	Bepalen van de effectiviteit van een filter .....	50
7.4	UML-diagram .....	51
8	Hardware.....	52
8.1	Gebruikte software.....	52
8.1.1	Modelsim SE 6.4a .....	52
8.1.2	ISE 9.1 .....	52
8.2	VERILOG & VHDL.....	52
8.2.1	Verilog .....	52
8.2.2	VHDL .....	52
8.2.3	Samenwerking VHDL & Verilog .....	53
8.3	Signed getallen .....	53
8.3.1	Conversie van binair naar 2's complement.....	54
8.4	FIR .....	54
8.5	LUT .....	54
8.6	Optimalisatie van hardwarecomponenten .....	57
8.7	Simulatie met de TEXTIO-library.....	58
8.8	Effectiviteit van het filter .....	59
8.9	Implementatie van de FIR-filter.....	60
8.9.1	Implementatie per QIE-kanaal .....	60
8.9.2	Implementatie per subtower .....	61
9	Triggerresultaten.....	64

10	Besluit.....	66
	Lijst van tabellen.....	67
	Lijst van figuren .....	68
	Bibliografie .....	70
	APPENDIX I:LVDS .....	73
	Appendix II: Opstellen van de lijnvergelijking .....	75
	APPENDIX III: afleiding reflectiecoëfficiënt .....	76
	APPENDIX IV: .....	77
	APPENDIX V: Zero padding.....	78

## **Gebruikte afkortingen**

ADC: Analoog-Digitaal Convertor  
CASTOR: Centauro And Strange Object Research  
CMS: Compact Muon Solenoid  
FIR: Finite Impulse Response  
FPGA: Field-programmable gate array  
GOL: Gigabit Optical Link  
HCAL: Hadron Calorimeter  
HDL: Hardware Description Language  
HTR: HCAL Trigger/ Read-out  
PMT: PhotoMultiplier Tube

# 1. Inleiding

De onderzoeksgroep elementaire deeltjesfysica aan de universiteit van Antwerpen participeert aan een experimenteel onderzoek naar botsingen van deeltjes in de krachtigste deeltjesversneller ter wereld namelijk de Large Hadron Collider. Zij richten hun onderzoek op het zoeken naar het Higgs-boson en het zoeken naar extra dimensies in de ruimte.

## 1.1 CERN

CERN is de afkorting van Conseil Européen pour la Recherche Nucléaire, vertaald Europese Raad voor Kernonderzoek. Later op 29 september 1954 werd de akte getekend die de raad ophief en verving door l'organisation européenne pour la Recherche nucléaire. Ondanks de akte blijft CERN als naam gebruikt worden.

De Europese organisatie voert experimenten uit om inzichten te verwerven in de opbouw van materie: uit welke deeltjes bestaat materie en welke krachten werken hierop in? Op dit moment wordt er veel onderzoek gedaan naar het bestaan van het Higgs-deeltje. Dat is een deeltje dat massa geeft aan andere deeltjes. Volgens het standaardmodel wegen alle elementaire deeltjes evenveel als licht: niets. Toch is het zo dat de elementaire deeltjes massa hebben. Peter Higgs zag hier als eerste een oplossing voor het massaprobleem, namelijk het Higgs-deeltje. Naast het onderzoek naar het Higgs-deeltje, zijn er nog een aantal andere experimenten aanwezig op het CERN.[5]

Het CERN is een internationale organisatie. Bij het ontstaan van de organisatie waren 12 landen lid. Sindsdien zijn er nog enkele landen toegetroten tot de organisatie. Op dit moment zijn er 20 landen lid van CERN.[3]

Oostenrijk	België	Bulgarije
Tsjechië	Denemarken	Finland
Frankrijk	Duitsland	Griekenland
Hongarije	Italië	Nederland
Noorwegen	Polen	Portugal
Slovenië	Spanje	Zweden
Zwitserland	Groot-Brittannië	

*Tabel 1: CERN leden*

Daarnaast zijn er ook enkele landen die het statuut van waarnemer hebben. Deze landen mogen aanwezig zijn op de bestuursvergaderingen en hebben het recht op het ontvangen van de hiervoor opgestelde stukken. Een waarnemer heeft geen stemrecht tijdens bestuursvergaderingen. Landen die het statuut van waarnemer aanvragen zijn meestal

landen dat geen lid kunnen of willen worden van CERN maar indien mogelijk wel willen deel nemen.

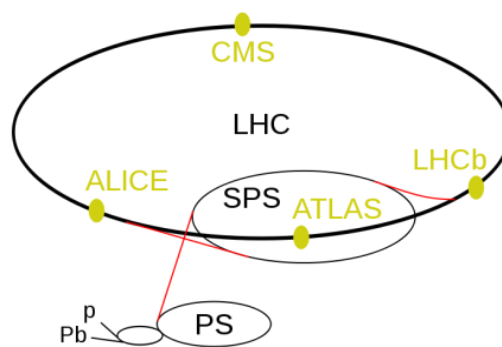
India	Israël	Japan
Rusland	Turkije	Verenigde Staten
UNESCO	Europese Commissie	

Tabel 2: CERN waarnemers

### 1.1.1 Large Hadron Collider of LHC

LHC is een gigantische deeltjesversneller gelegen nabij Genève.[6] Het is het grootste apparaat gemaakt door de mens tot nu toe. De LHC ligt in een tunnel 100m onder de grond en heeft een omtrek van 27km. De deeltjesversneller wordt gebruikt om elementaire deeltjes te bestuderen. Deze studie moet een betere kijk werpen op de structuur van atomen en het ontstaan van het universum.

Voorlopig is de LHC de krachtigste versneller op de planeet maar er zijn al plannen voor een krachtigere deeltjesversnellers. Bijvoorbeeld het ILC (International Linear Collider).



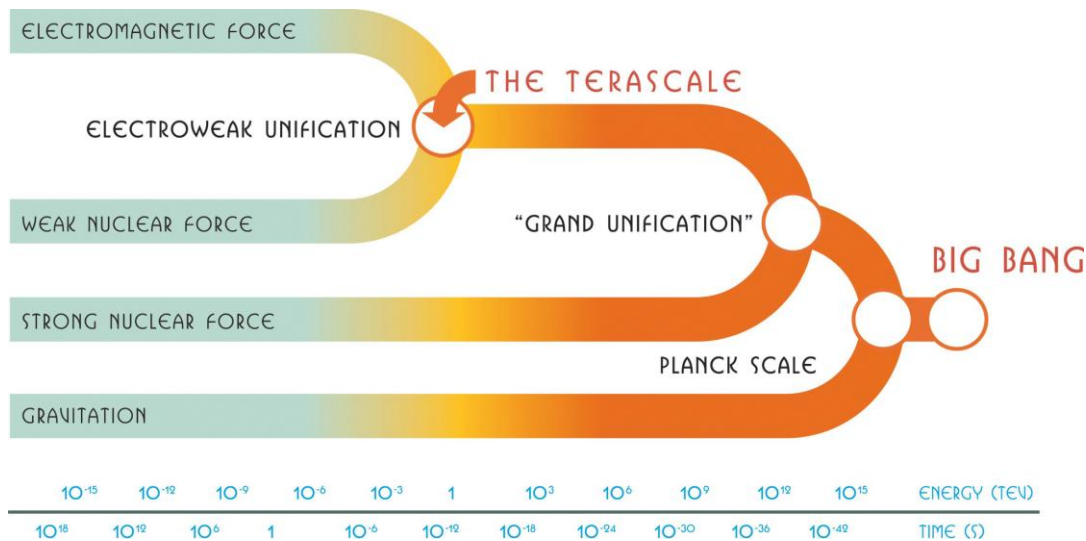
Figuur 1: LHC

### Wat gebeurt er?

In de LHC wordt door middel van supergeleidende elektromagneten de energie van deeltjes verhoogd tot hun snelheid 99,999 996 4% van de lichtsnelheid<sup>1</sup> bedraagt. In de deeltjesversneller worden twee stralen van deze deeltjes opgewekt die elk in een aparte pijp en in tegengestelde zin in de LHC bewegen. Wanneer de energie van de stralen groot genoeg is, laat men de stralen met elkaar botsen. Op de LHC zijn verschillende soorten detectoren aangebracht die informatie verzamelen over de botsingen. Er worden waterstofionen versneld in beide richtingen tot ze een energie van 7TeV bereiken, samen wordt een energiewaarde van 14TeV verkregen.

<sup>1</sup> Lichtsnelheid  $c = 299\,792\,458\text{ m/s}$

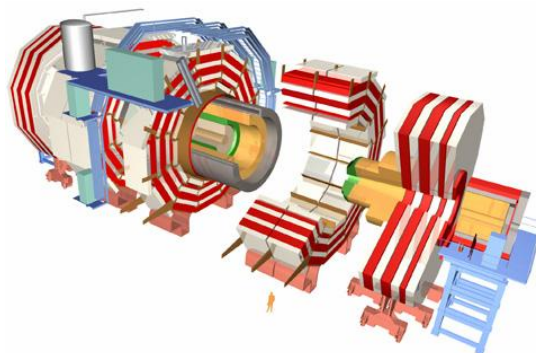
In de toekomst wordt gepland om loodionen met elkaar te laten botsen. Deze deeltjes zorgen voor een veel grotere energie, namelijk  $2\text{PeV}^2$ . Deze grotere energie is nodig om als het ware terug in de tijd te gaan naar de toestand van het universum net na de Big Bang. Dit wordt voorgesteld op onderstaand figuur.



*Figuur 2: Big Bang*

### 1.1.2 CMS of Compact Muon Solenoid

De Compact Muon Solenoid is een van vier detectoren aanwezig op de LHC. Het is een van de twee algemene detectoren die rond een interactiepunt van de LHC geplaatst zijn. De detector bestaat uit verschillende lagen die elk specifieke eigenschappen bezitten voor het opvangen en het meten van de energie en het moment van de deeltjes. Hoe hoger het moment van een geladen deeltje, hoe minder het deeltje zal afbuigen in een magnetisch veld. Het eerste element dat zeer belangrijk is voor de CMS-detector[4] is een sterke magneet. Een sterke magneet bepaalt de resolutie van de energiemeting van de deeltjes.

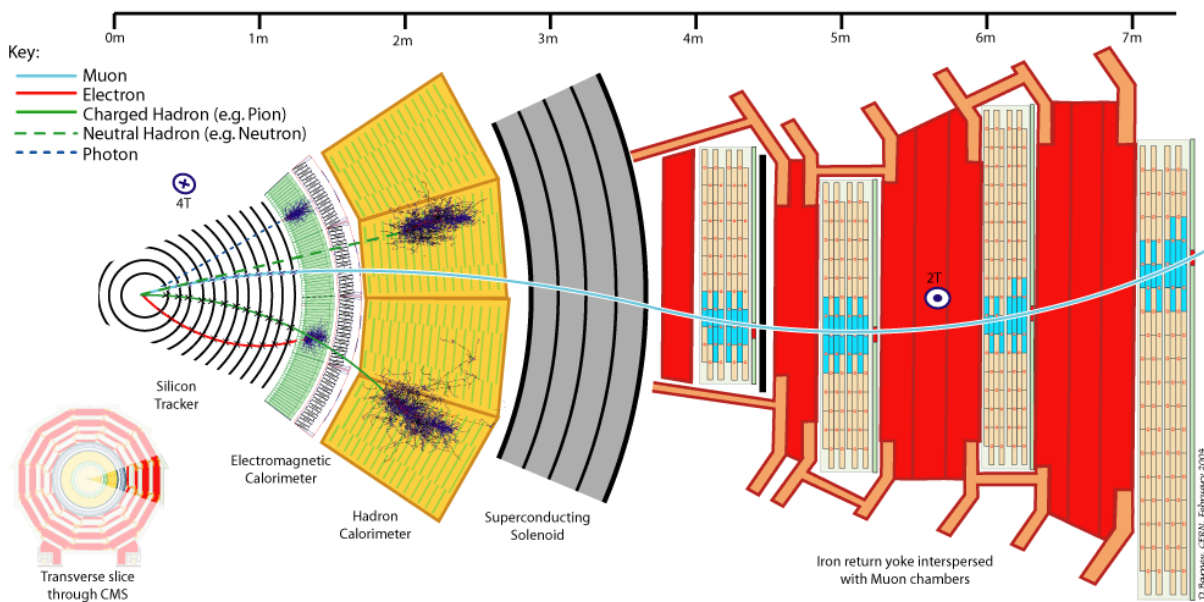


*Figuur 3: CMS detector*

<sup>2</sup> P staat voor Peta en is equivalent aan  $10^{15}$

## Structuur

Het CMS heeft een cilindervormige structuur. Het bestaat uit verschillende detectoren die concentrisch opgebouwd zijn rond het interactiepunt<sup>3</sup>.



Figuur 4: Doorsnede CMS-detector

Hierboven een dwarsdoorsnede van de CMS detector:

- Puntendetector (Pixel detector)
- Sporenkamer (Silicon tracker)
- Elektromagnetische calorimeter
- Sampling-hadronische calorimeter
- Hadronische calorimeter
- Muonkamers

De puntendetector bestaat uit drie lagen die net als een sensor van een fotoapparaat beeldpunten hebben. De belangrijkste taak van deze detector is het bepalen van het interactiepunt in micrometer precisie.

De sporenkamer bestaat uit zeven lagen silicium strip detectoren. Een silicium strip detector is een silicium schijf met een oppervlakte van tientallen vierkante centimeter en een dikte van enkele honderden micrometer. De bedoeling van de sporenkamer is om de trajecten van de deeltjes zo nauwkeurig mogelijk te reconstrueren. Het is hierbij belangrijk dat de sporenkamer het traject van de deeltjes zo weinig mogelijk beïnvloedt. Het materiaal van de sporenkamer is zo gekozen dat de deeltjes niet kunnen botsen en niet van richting kunnen veranderen. Om het traject van een deeltje vast te leggen wordt meermaals de positie van

<sup>3</sup> Dit is het punt waar de botsing optreedt

het deeltje gemeten zodat een reconstructie van het traject nauwkeurig is tot op  $10\mu\text{m}$ . Wanneer een deeltje door de siliciumplaten gaat, wordt er een elektron geëxciteerd<sup>4</sup> en ontstaat er een elektron-gatpaar. Door een spanningsverschil aan te leggen over de siliciumplaten zal ten gevolge van deze elektron-gatparen een stroom ontstaan in het materiaal van de detector. Deze stroom vormt een elektrisch signaal. Dit signaal wordt uitgelezen door een computer en wordt omgezet naar een 'spoor'. De magneet van het CMS zorgt voor een sterk magnetisch veld dat de deeltjes in de sporenkamer doet afbuigen. Aan de hand van de kromtestraal van de gevolgde baan wordt de impuls van het deeltje berekend.

Na de sporenkamer komen de deeltjes terecht in de calorimeters. De calorimeters zijn ontworpen om nauwkeurig de energie van deeltjes te meten. Ze bestaan uit dicht materiaal, zodat de deeltjes vrijwel onmiddellijk botsen. Wanneer er een elementair deeltje met zeer hoge energie met het materiaal botst, ontstaat een lawine nieuw gevormde elementaire deeltjes. Er worden verschillende technieken gebruikt om deze lawine van deeltjes om te zetten in lichtpulsen. Vervolgens meet een fotodiode de hoeveelheid uitgezonden licht. Deze hoeveelheid licht is een maat voor de energie van de gebotste deeltjes.

De meeste deeltjes worden geabsorbeerd in de calorimeters maar deeltjes zoals muonen en neutrino's gaan er dwars doorheen. Om de muonen toch te kunnen detecteren worden er opnieuw sporenkamers rond de calorimeters geïnstalleerd. Deze constructie wordt muonkamer genoemd. De neutrino's kunnen niet worden gedetecteerd. De aanwezigheid van neutrino's wordt aangetoond doordat de energiebalans niet klopt. De energiebalans is de energie die bij de botsing vrijkomt en de totale energie van alle deeltjes die worden gedetecteerd.

## 1.2 CASTOR

CASTOR is de afkorting van Centauro And Strange Object Research.[19] De calorimeter werd mede ontwikkeld door de universiteit van Antwerpen. Hij maakt deel uit van de CMS-detector en ligt mee op de LHC. De detector is gepositioneerd in de lengte van de protonenbundel op 14,38 meter van het interactiepunt. De CASTOR detector is verantwoordelijk voor het meten van de energie van deeltjes na proton-protonbotsingen.

---

<sup>4</sup> Excitatie gebeurt wanneer er genoeg energie wordt toegevoegd aan een elektron zodanig dat het zich van de kern van het atoom kan verwijderen.

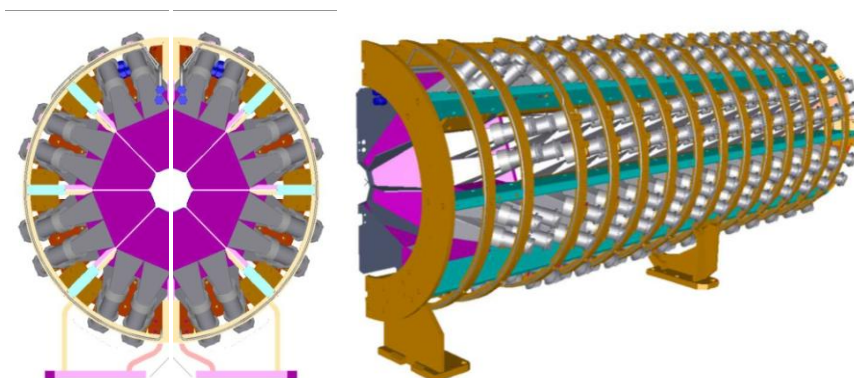


*Figuur 5: CASTOR*

Hij is opgebouwd uit afwisselend detectie- en absorptieplaten zoals bij een standaard calorimeter. Het absorberend materiaal is wolfram. De detectieplaten bestaan uit kwartskristal. Wanneer een invallend deeltje met wolfram botst, ontstaat er een explosie van elementaire deeltjes.

Deze geladen deeltjes gaan door het kwartskristal. Het passeren van de deeltjes gebeurt tegen de lichtsnelheid. Indien de snelheid groter is dan de maximale snelheid in het medium dan wordt er energie afgestaan door fotonen uit te zenden. Dit fenomeen wordt het Cherenkov-effect genoemd.[13] De fotonen worden vervolgens gedetecteerd en het signaal wordt versterkt door een fotomultiplier. Vervolgens wordt het signaal verzonden naar een uitleeskaart (QIE).

CASTOR is ingedeeld in 16 sectoren. Elke sector is opgebouwd uit 14 longitudinale modules. Elke module heeft zijn eigen uitleeskanaal: de PMT. Er zijn in het totaal dus 224 PMT's (16x14).

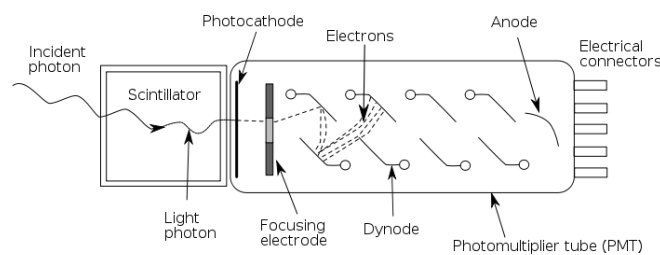


*Figuur 6: CASTOR voor- en zijaanzicht*

Op vorige figuur wordt de structuur van CASTOR weergegeven. Indien er gekeken wordt naar het vooraanzicht van CASTOR, kan gezien worden dat CASTOR 16 sectoren bezit. Een sector bestaat uit 14 longitudinale PMT-modules m.a.w. de PMT's van één sector liggen in elkaars verlengde. De 14 longitudinale modules worden geïllustreerd in het zijaanzicht.

### 1.3 PMT

PMT is de afkorting voor photomultiplier tube. Een PMT zorgt voor de omzetting van lichtenergie in een elektrisch signaal. Hiervoor maakt het gebruik van een anode<sup>5</sup>, fotokathode<sup>6</sup> en enkele dynodes. Wanneer een foton invalt op de fotokathode, produceert deze een elektron. Dit effect wordt het foto-elektrisch effect genoemd.[18]



*Figuur 7: De werking van een PMT*

In de tube zitten verschillende dynodes. Door de plaatsing van een elektrode worden de elektronen gefocusseerd in de richting van een dynode. Het gegenereerde elektron beweegt zich naar een positief geladen dynode. De positieve spanning van de dynode zorgt voor een versnelling van het elektron.

Wanneer de eerste dynode bereikt wordt, worden er meerdere elektronen met lagere energie geëmitteerd. De elektronen verplaatsen zich van dynode naar dynode omdat de volgende dynode telkens een iets hoger positieve spanning heeft. Ten slotte bereiken de elektronen de anode. Hier zorgt de verzameling elektronen voor een pulsvormig stroomverloop.

Het Cherenkov effect is een direct effect en het passeren van het deeltje in de quartz plaat is van zeer korte duur (ps). Dit resulteert in een stroom van fotonen die slechts enkele ns duurt. Het stroomsignaal uit de PMT zal korter zijn dan 10ns.

<sup>5</sup> Anode: de pool naar waar de elektronen zich bewegen

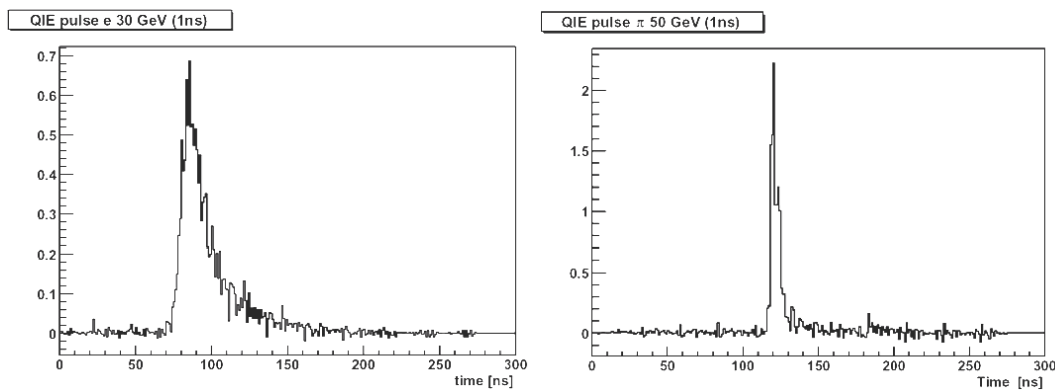
<sup>6</sup> Kathode: de elektrode waaruit de elektronen komen

## 1.4 QIE

De PMT verstuurt een stroomsignaal  $i(t)$  over de coaxiale kabel naar de QIE-kaart. QIE staat voor Charge[Q], Integration en Encoding. Dit signaal wordt geïntegreerd in de tijd over 25ns en vervolgens gesampled. Door de integratie wordt van het stroomsignaal een spanning gemaakt aan de hand van volgende formule:

$$U(t) = \int_0^{25ns} \frac{i(t)}{C} dt$$

Het stroomsignaal wordt aangelegd aan een condensator. Deze wordt ontladen via een amplitudeafhankelijke impedantie. Deze impedantie zorgt ervoor dat de pulsvorming sneller is bij grote signalen dan bij kleine signalen. Het gebruik van de amplitudeafhankelijke impedantie kan bijdragen tot het ontstaan van de signaalstaart.



*Figuur 8: QIE pulsvorming voor klein en groot signaal*

Ondanks het feit dat de PMT pulsen genereert van  $\pm 10$ ns, worden er in praktijk pulsen waargenomen die breder zijn dan 25ns. Een deel van de puls wordt dus verwerkt tijdens een volgende integratie. Zo zit de informatie van één puls vervat in twee of meerdere integraties.

De volgende stap is het digitaliseren van de data. Het aangelegde analoge signaal wordt na integratie door middel van twee comparators/vergelijkers ingedeeld in vier verschillende bereiken. Dit bereik zal het niet-lineaire karakter van de niet-lineaire ADC bepalen.

Onderstaande tabel geeft het niet-lineaire karakter weer. De range of exponent wordt bepaald door de comparators. De comparator bepaalt met welke versterking het signaal wordt aangeboden aan de ADC. Wanneer bijvoorbeeld het aangelegde signaal overeen komt met een lading van 30 fC, dan zullen de comparators het signaal classificeren als range 0.

**Non-Inverting Input Scale (PMT inputs)**

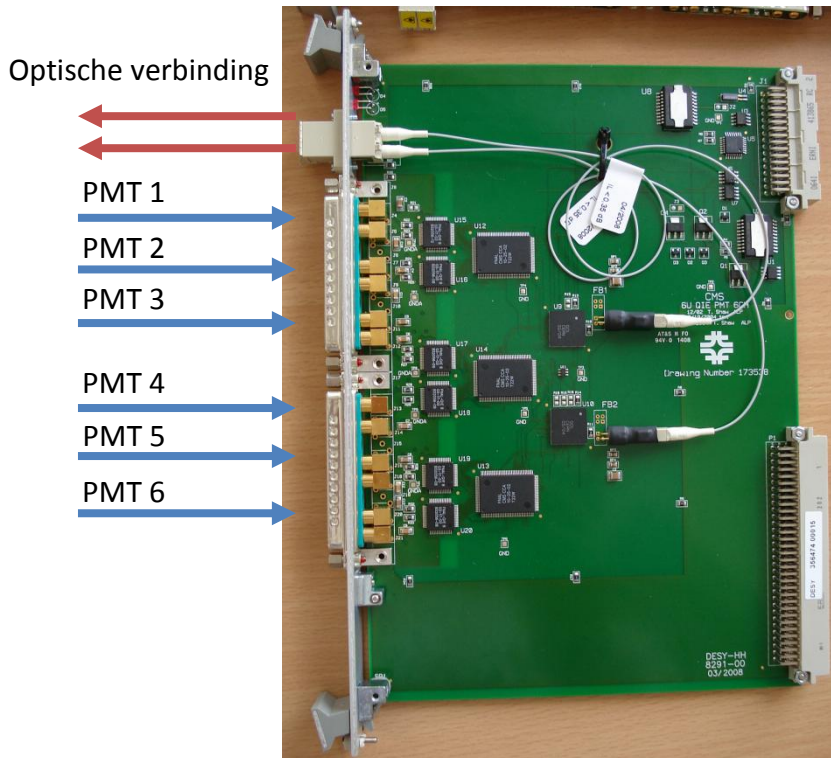
Normal Mode			
Range (Exponent)	Input Charge	FADC Codes	Gain (q/Lsb)
0	-3 fC --- 36 fC	0---14	2.6 fC/bin
0	36 fC --- 73 fC	15---21	5.2 fC/bin
0	73 fC --- 104 fC	22---25	7.8 fC/bin
0	104 fC --- 135 fC	26---28	10.4 fC/bin
0	135 fC --- 174 fC	29---31	13 fC/bin
1	150 fC --- 343 fC	0---14	13 fC/bin
1	343 fC --- 525 fC	15---21	26 fC/bin
1	525 fC --- 681 fC	22---25	39 fC/bin
1	681 fC --- 837 fC	26---28	52 fC/bin
1	837 fC --- 1032 fC	29---31	65 fC/bin
2	902 fC --- 1877 fC	0---14	65 fC/bin
2	1877 fC --- 2787 fC	15---21	130 fC/bin
2	2787 fC --- 3567 fC	22---25	195 fC/bin
2	3567 fC --- 4347 fC	26---28	260 fC/bin
2	4347 fC --- 5322 fC	29---31	325 fC/bin
3	4672 fC --- 9547 fC	0---14	325 fC/bin
3	9547 fC --- 14097 fC	15---21	650 fC/bin
3	14097 fC --- 17997 fC	22---25	975 fC/bin
3	17997 fC --- 21897 fC	26---28	1300 fC/bin
3	21897 fC --- 26772 fC	29---31	1625 fC/bin

*Figuur 9: Conversietabel niet-lineaire ADC [22]*

Vervolgens wordt de spanning aangelegd aan een 5 bit niet-lineaire ADC met een conversie zoals aangegeven in de tabel onder de FADC kolom.

Nu wordt de 5 bit ADC-data en 2 bit range-data geserialiseerd tot 7 bit QIE-data. Hierbij wordt de exponent voorgesteld door 2 MSB (Most Significant Bit).

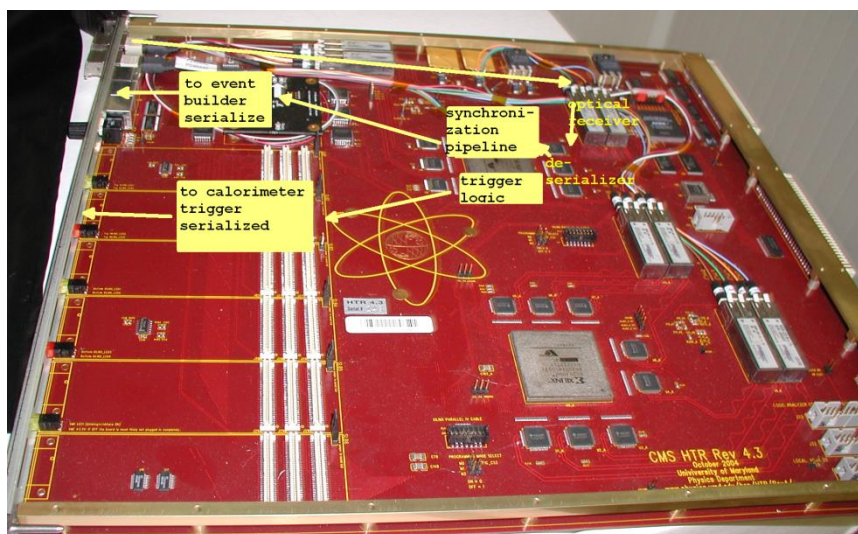
Tenslotte wordt de output van 3 QIE-kanalen tezamen met 11 bit controledata gecombineerd tot een 32 bit datawoord. Deze 32 bit data wordt over een Gigabit Optical Link (GOL) gestuurd met een snelheid van 40MHz. De GOL wordt verbonden met een HCAL Trigger/ Read-out kaart of kortweg HTR-kaart. Per QIE-kaart zijn er zes QIE-kanalen aanwezig. Dit resulteert in twee optische verbindingen per kaart.



Figuur 10: QIE-kaart

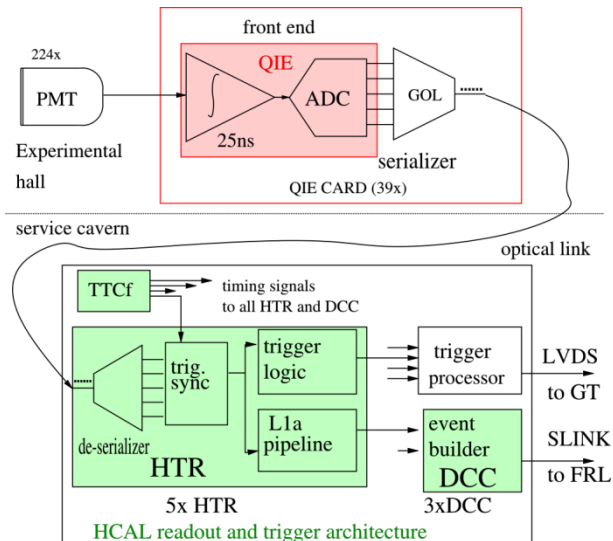
### 1.5 CASTOR read-out en triggerarchitectuur / data acquisitie hardware

Een HTR-kaart verzamelt 48 QIE-kanalen via 16 GOL's afkomstig van 8 QIE-kaarten. De kaart is verantwoordelijk voor het uitlezen en bewerken van data en voor het genereren van een Level-1 Trigger primitives. Dit zijn signalen dat aan hiërarch hogere hardware duidelijk maakt dat CASTOR interessante data bevat.



Figuur 11: HTR-kaart

Een HTR-kaart bestaat uit twee identieke helften. Elke helft bevat één FPGA dat 24 QIE-kanalen verwerkt. Per FPGA komt er elke 25ns data binnen van 24 QIE-kanalen. Elke FPGA kan zijn eigen triggerprimitieven sturen op basis van de verkregen QIE-data. De signaalstarten van de QIE-data veroorzaken foutieve triggerprimitieven. Er zal getracht worden de signaalstarten zo goed mogelijk te onderdrukken/corrigeren door gebruik te maken van digitale filters. (Zie hoofdstuk 6: FIR )



Figuur 12: Data path

Bij het genereren van een trigger<sup>7</sup>, wordt niet alleen de data van CASTOR gebruikt maar ook die van andere CMS calorimeters en muon detectoren. Indien er interessante data wordt vastgesteld, wordt een L1A(ccept) teruggestuurd naar de uitleeskaarten via het Timing, Trigger and Control (TTC) systeem. De volledige triggerarchitectuur krijgt 3,2  $\mu$ s om een beslissing te maken of de data interessant is.[19] De triggerprocessor maakt gebruik van de technologie LVDS, voor meer informatie raadpleeg appendix I.

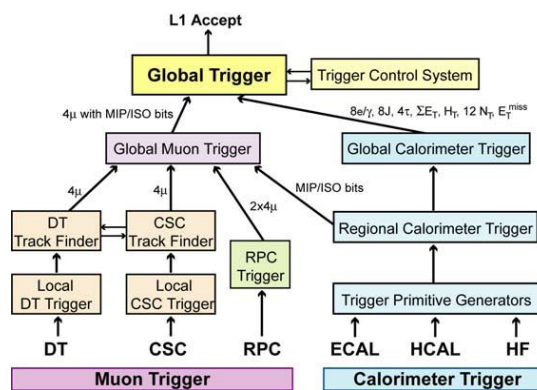
De data die geselecteerd wordt door een trigger, heet event-data. Deze event-data wordt doorgestuurd naar de Data Concentrator Card (DCC). Hier zal de data van alle kanalen door een eventbuilder worden gecombineerd tot een dataframe met een footer en een header. Vervolgens wordt de data doorgestuurd naar de Data Acquisition (DAQ). De CMS DAQ architectuur kan vergeleken worden met een grote dataswitch. De eventdata van de verschillende bronnen wordt via een switch gestuurd naar een van 500 verwerkingseenheden (computerfarm). Hier gebeurt een eerste analysestap, namelijk de High Level Trigger (HLT). Ook hier zal de data gecorrigeerd moeten worden. De signaalcorrectie kan hier plaatsvinden door een softwarealgoritme.[7]

<sup>7</sup> Trigger is een elektronische term dat het best kan worden omschreven als een signaal dat andere logica kan activeren wanneer er aan bepaalde grenswaarden voldaan wordt.

Er moet opgemerkt worden dat L1A een stochastische signaal is daar de verschillende botsingen onafhankelijk van elkaar zijn. Het doel van de Level-1 Trigger (L1T) is om de data rate te herleiden tot 100 kHz. Daar het niet mogelijk om de data van alle botsingen te bewaren of te verwerken.

Op volgende figuur wordt de architectuur van de L1T afgebeeld. De L1T heeft drie type componenten: locale, regionale en globale triggerinformatie. De locale triggers zijn afkomstig van gemeten energie in de calorimeters.

Regionale triggers maken gebruik van triggerinformatie en patronen om een trigger te genereren. Ten slotte is er de globale trigger. Deze trigger maakt een beslissing op basis van berekeningen via algoritmes die in een FPGA zijn geïmplementeerd.

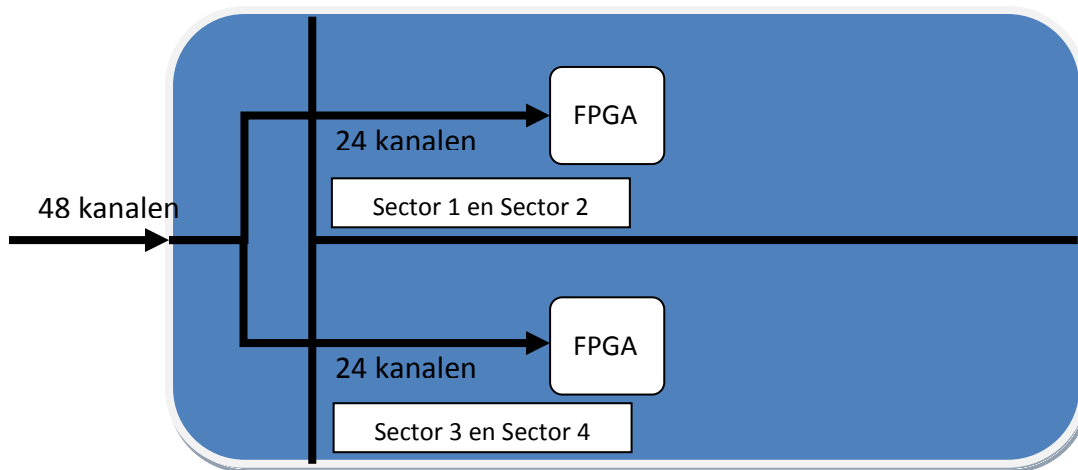


Figuur 13: Triggerarchitectuur

## 1.6 De triggerlogica

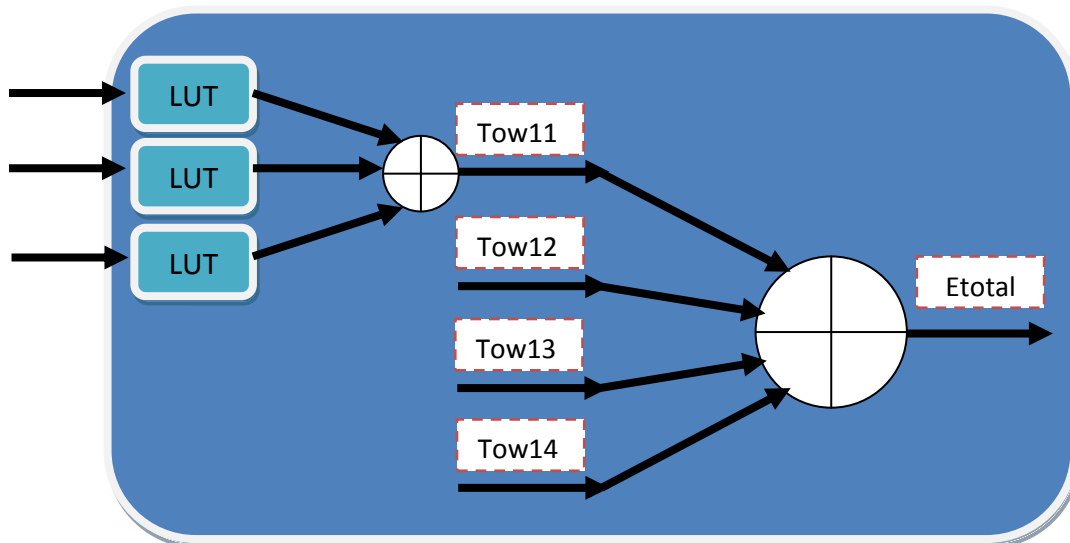
Een HTR-kaart bezit twee identieke FPGA's die instaan voor het verwerken van data en het genereren van triggerprimitieven. Om deze functionaliteit te bekomen, zijn de FPGA's geprogrammeerd met de nodige logica. De belangrijkste logische component voor het genereren van triggerprimitieven is het TriggerPath.

QIE-data van twee sectoren van CASTOR worden doorgestuurd naar het TriggerPath. Per sector heeft CASTOR 14 modules (PMT's). De 12 eerste modules worden doorgestuurd naar dezelfde HTR-kaart. De data van de laatste twee modules worden naar een andere HTR-kaart gestuurd. Deze worden niet gebruikt voor het TriggerPath maar wel voor het HLT-algoritme. Een volledige HTR-kaart verwerkt dus vier sectoren van CASTOR.



*Figuur 14: Structuur HTR-kaart*

Eerst wordt de data van de 24 QIE-kanalen gelineariseerd door middel van een LUT<sup>8</sup>. Het TriggerPath bevat 24 LUT, een voor elk kanaal. Deze parallelle LUT's zijn nodig om al de kanalen gelijktijdig te lineariseren.



*Figuur 15: Structuur TriggerPath*

Per sector wordt er een triggertower voorzien. De energie van drie aangrenzende modules wordt gesommeerd. De som wordt toegewezen aan een subtower. Zo bevat tow11 de energie van module 1 tem 3. De volgende subtower bevat telkens de energie van de drie volgende modules.

<sup>8</sup> Look-up table

De tower bestaat uit vier subtowers: tow11, tow12, tow13, tow14 voor de eerste sector. Voor de tweede sector zijn de subtowers: tow21, tow22, tow23, tow24

Tenslotte wordt de totale energie-inhoud van een sector berekend door de som van de subtowers te nemen. De totale energie-inhoud van een sector zal vergeleken worden met een bepaalde drempelwaarde om vervolgens een triggerprimitief te genereren.

Er zijn drie soorten triggers aanwezig op de HTR-kaart:

1. Energytrigger: hier wordt de totale energie-inhoud van een sector vergeleken met een bepaalde drempelwaarde. Indien de energie-inhoud groter is dan de drempelwaarde, zal er een trigger gegenereerd worden.
2. NolevelTrigger: deze trigger gaat kijken of er energie aanwezig is in de eerste zes modules van een sector. Hiervoor wordt tow11 met tow12 gesommeerd. Indien de energie-inhoud kleiner is dan een opgegeven drempelwaarde, zal er een trigger aangemaakt worden.
3. MuonTrigger: dit is een complexere trigger. De energie van elke subtower moet kleiner zijn dan een bepaalde bovengrens maar moet ook groter zijn dan een bepaalde noise-marge. Indien alle subtowers voldoen aan de specificatie, dan wordt een triggersignaal aangemaakt.

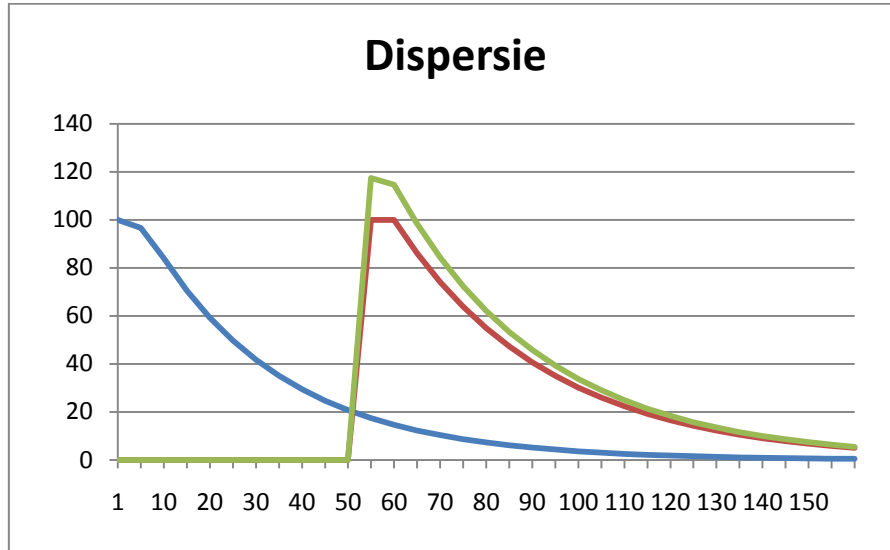
Een trigger wordt gegenereerd indien de energie-inhoud van een bepaalde sector of subtower voldoet aan bepaalde voorwaarden. De energie van de QIE-staarten heeft vaak genoeg amplitude om triggers zoals de energytrigger en de Noleveltrigger te induceren zonder dat daar een botsing mee verband houdt. Om deze foutieve triggers tegen te gaan, moet er een correctie plaatsvinden op de energie-inhoud van de verschillende modules van CASTOR. De correctie kan doorgevoerd worden op twee niveaus, namelijk op QIE-niveau of op subtowerniveau. Op QIE-niveau zal de energie van elke QIE-kanaal individueel gecorrigeerd worden. Op subtowerniveau zal er per drie QIE-kanalen een correctie plaatsvinden. Waar de correctie kan doorgevoerd worden, wordt meer in detail besproken in hoofdstuk 6: FIR filter.

## 2 Het probleem

### 2.1 De probleemstelling

In CASTOR worden botsingen gedetecteerd. Door het Cherenkov-effect ontstaat er een lichtflits. Dit licht wordt opgevangen voor licht-geleiders en wordt overgebracht naar fotomultipliers (zie hoofdstuk 1.3: PMT). Het stroomsignaal wordt naar een QIE-kaart (zie hoofdstuk 1.4: QIE) gestuurd over een transmissielijn van 12m. Deze transmissielijn zorgt voor vervorming van het signaal. Deze vervorming kan veroorzaakt worden door dispersie, ohmse verliezen en het inductieve en capacitieve karakter van de transmissielijn.

De stroom wordt met behulp van een ADC omgezet in een digitaal signaal. Als gevolg van dispersie zijn de pulsen langer geworden dan 25ns waardoor de informatie van één botsing uitgesmeerd wordt over meerdere integratie-intervallen. Op de transmissielijn is dispersie een probleem aangezien de QIE-kaart het signaal iedere 25ns integreert en er iedere 25ns een nieuwe botsing kan gebeuren. De signalen kunnen als gevolg van dispersie overlappen (zie onderstaande grafiek). Daarnaast maakt de QIE-kaart gebruik van een amplitudeafhankelijk impedantie. Zowel de dispersie als de impedantie kunnen zorgen voor het ontstaan van de staart van het signaal. Deze zorgt voor foutieve trigger events (zie hoofdstuk 1.5 CASTOR read-out en hoofdstuk 1.6: triggerlogica).



### 2.2 De opdracht

In deze thesis zal onderzocht worden in welke mate de transmissielijn verantwoordelijk is voor de ontstane signaalstaarten. Daarvoor is een uitvoerig onderzoek van de coaxiale transmissielijn nodig. Er zal onderzocht worden hoe een transmissielijn gemodelleerd kan worden, hoe de transferfunctie opgesteld kan worden, welke verliezen aanwezig zijn op de

lijn, fasesnelheid, dispersie en reflecties. Het doel van dit onderzoek is het gedrag van de coaxiale kabel zo goed mogelijk te kunnen voorspellen.

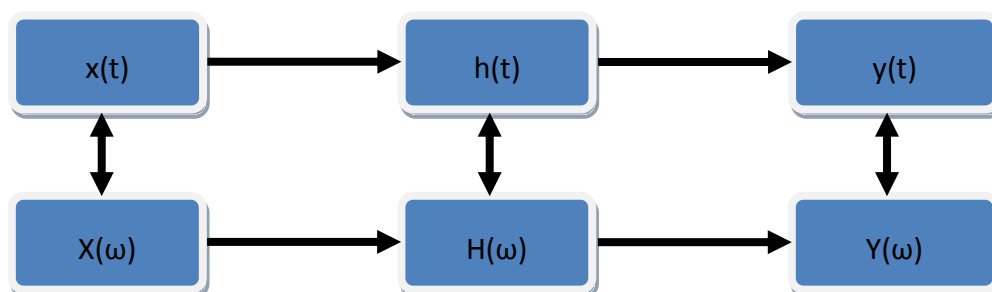
Vervolgens moeten de signaalstaarten onderdrukt worden. Hiervoor zal gebruik gemaakt worden van digitale filters, namelijk FIR filters. De specificatie en de implementatie van de filter zullen onderzocht moeten worden om een zo optimaal mogelijk resultaat te verkrijgen. Het doel is om de firmware van de HTR-kaarten aan te passen zodat de signaalstaarten geen invloed meer hebben op triggers.

### 2.3 Het plan van aanpak

Op voorwaarde dat het ontstaan van de signaalstaarten veroorzaakt wordt door de transmissielijn, moet de mogelijkheid bestaan om dit gedrag te simuleren/voorspellen aan de hand van een model. Hiervoor bestaan twee manieren van aanpak. Enerzijds kunnen er metingen gedaan worden op de coaxiale kabel. Deze metingen kunnen gebruikt worden om het gedrag mathematisch vast te leggen. Anderzijds kan er een mathematisch model opgesteld worden om het gedrag te simuleren. Om dit te realiseren is er een studie van de coaxiale kabel nodig (Zie hoofdstuk 3: Transmissielijn).

Indien er een model voor de transmissielijn gevonden wordt, kan aan de hand van dit model de specificatie van een FIR filter worden bepaald. De specificatie van de filter wordt afgeleid aan de hand van volgende Figuur 16: Transformatieschema.

Als het model van de transmissielijn niet voldoet aan de verwachtingen, zijnde de signaalstaarten kunnen simuleren, dan wordt er een signaalanalyse toegepast op de QIE-data. Aan de hand van die signaalanalyse is het mogelijk om een gemiddelde impulsresponsie op te stellen. Deze impulsresponsie zal vervolgens gebruikt worden bij het opstellen van de filterparameters. Deze bepalen de responsie van een filter. Ten slotte wordt deze filter geschreven in VHDL en wordt deze geïmplementeerd in de firmware van de HTR-kaarten.

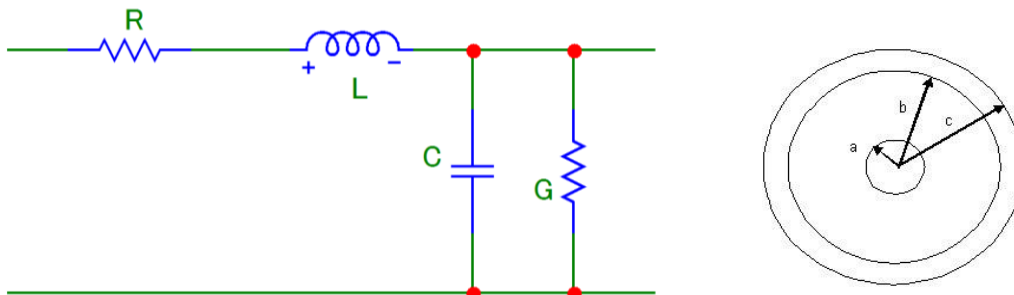


*Figuur 16: Transformatieschema*

### 3 Transmissielijn

Voor de analyse van een transmissielijn kan deze worden voorgesteld aan de hand van volgende componenten:

- R: serie weerstand,  $\Omega/m$
- L: serie inductantie, H/m
- C: parallel capaciteit, F/m
- G: parallel weerstand,  $1/\Omega m$



Figuur 17: Model

Bovenstaande figuur is een voorstelling van een elementair deel van een transmissielijn. Per deel kan geschreven worden:[12]

$$\frac{\delta V}{dz} = (R + j\omega L)\delta I \quad (3.1)$$

$$\frac{\delta I}{dz} = -(G + j\omega C)\delta V \quad (3.2)$$

Indien bovenstaande vergelijkingen opgelost worden, kan een signaal worden voorgesteld op de volgende manier:  $V=V_0e^{-\gamma x}$  met propagatiesnelheid  $\gamma$  en afstand  $x$ . Vervolgens kunnen volgende formules worden afgeleid:

$$Z_0 = \frac{V}{I} = \sqrt{\frac{R + j\omega L}{G + j\omega C}} \sim \sqrt{\frac{L}{C}} \quad (3.3)$$

$$\gamma = \sqrt{(R + j\omega L)(G + j\omega C)} \sim j\omega\sqrt{LC} \quad (3.4)$$

Bij hoge frequentie kunnen de formules 3.3 en 3.4 vereenvoudigd worden door R en G te verwaarlozen.

De propagatiesnelheid  $\gamma$  is een complex getal en kan geschreven worden als  $\gamma = \alpha + j\beta$ . Hierbij is  $\alpha$  de dempingsfactor en  $\beta$  de fasefactor. Vaak worden deze termen constant verondersteld maar eigenlijk is dat niet van toepassing aangezien beide parameters afhangen van de frequentie. Daarnaast is het ook zo dat de parameters R, G en L zelf afhankelijk zijn van de frequentie. Zo is bijvoorbeeld R afhankelijk van de frequentie als gevolg van het skineffect. De parameter C is afhankelijk van het gebruikte isolatiemateriaal en de vorm van de kabel. Wanneer formule 3.4 verder wordt uitwerkt naar een reëel en imaginair deel, worden onderstaande formules verkregen:

$$\alpha = \sqrt{\frac{1}{2} \left( (RG - \omega^2 LC) + \sqrt{(R^2 + \omega^2 L^2)(G^2 + \omega^2 C^2)} \right)} \quad (3.5)$$

$$\beta = \sqrt{\frac{1}{2} \left( \sqrt{(R^2 - \omega^2 L^2)(G^2 + \omega^2 C^2)} - (RG - \omega^2 LC) \right)} \quad (3.6)$$

De afleiding van bovenstaande formules werd opgenomen in appendix II.

Er bestaan twee manieren waarop de verliezen van een transmissielijn in kaart kunnen gebracht worden. Er kan naar de frequentieafhankelijkheid van elke parameter gekeken worden. Zo neemt bijvoorbeeld R toe naarmate de frequentie stijgt als het gevolg van het skineffect<sup>9</sup>.

Een tweede manier is om te onderzoeken welke verschillende verliezen er mogelijk zijn op een transmissielijn, in dit geval een coaxiale kabel. De demping op een transmissielijn kan opgedeeld worden in vier verliezen:

- Verliezen afhankelijk van de geleidbaarheid  $\alpha_M$  ( inclusief het skineffect)
- Verliezen afhankelijk van de diëlektrische verliestangens  $\alpha_D$
- Verliezen afhankelijk van de geleidbaarheid van het diëlektricum  $\alpha_G$
- Verliezen door straling  $\alpha_R$ : verwaarloosbaar

Samen vormen ze de totale demping  $\alpha = \alpha_M + \alpha_D + \alpha_G$

Het grootste deel van de verliezen op een transmissielijn gebeurt als gevolg van ohmse verliezen. Deze verliezen worden gemodelleerd door de component R in het transmissielijn model. R is afhankelijk van de geometrie van het materiaal en van de RF sheet resistance. Dit is de weerstand van de oppervlakte van de coaxiale kabel.

---

<sup>9</sup> Bij hoge frequenties wordt de stroom naar de buitenkant van de geleider geduwd, dit verhoogt de weerstand.

$$\text{Inner conductor resistance/length} = \frac{\rho_{inner}}{\text{Area}_{inner}} = \frac{\rho_{inner}}{2\pi\delta \left[ a + \delta \left( e^{\frac{-a}{\delta}} - 1 \right) \right]} \quad (3.7)$$

$$\text{Outer conductor resistance/length} = \frac{\rho_{outer}}{\text{Area}_{outer}} = \frac{\rho_{outer}}{2\pi\delta \left[ b + \delta \left( e^{\frac{-b}{\delta}} - 1 \right) \right]} \quad (3.8)$$

De skindiepte kan berekend worden aan de hand van volgende formule:  $\delta = \sqrt{\frac{2\rho}{\omega\mu}}$ .

Bij een stijging van de frequentie, daalt de skindiepte en dus ook de beschikbare oppervlakte waardoor er stroom kan vloeien. Omdat de soortelijke weerstand constant is, zullen beide weerstanden stijgen bij een frequentiestijging.

Vooraleer er verder gegaan wordt met deze beschouwing, moet er een woordje uitleg gegeven worden over de eenheid Neper<sup>10</sup>. Neper is een logaritmische eenheid zonder dimensie dat gebruikt wordt in de wereld van elektronische signalen om een versterking of een verzwakking uit te drukken.

De logaritmische schaal van Neper maakt gebruik van het natuurlijke logaritme LN, gebaseerd op het getal van Euler e (=2,71828). Met volgende formule wordt de demping bekomen in Neper.  $v_1$  en  $v_2$  zijn twee spanningen op een bepaalde afstand van elkaar.

$$L_{Np} = \ln \frac{v_1}{v_2} = \ln v_1 - \ln v_2$$

Daar er een relatie bestaat tussen de verschillende logaritmische schalen, kan Neper uitgedrukt worden in dB.

$$1 \text{ Np} = \frac{20}{\ln 10} \text{ dB} = 20 \log_{10} e \text{ dB} = 8,686 \text{ dB}$$

De weerstand van de kern en de mantel mogen met elkaar opgeteld worden. Zo wordt de totale weerstand/lengte verkregen. Deze totale weerstand moet vervolgens gedeeld worden door twee maal de karakteristieke impedantie. [14]

$$\alpha_M = \frac{\text{Totale weerstand / lengte}}{2Z_0} [\text{Np/m}] \quad (3.9)$$

Demping door diëlektrische verliezen wordt dominantier naarmate de frequentie van het signaal stijgt. Algemeen kan de formule geschreven worden op de volgende manier:

$$\alpha_D = \tan(\delta) \frac{\omega CZ_0}{2} [\text{Np/m}] \quad (3.10)$$

<sup>10</sup> Zowel neper als decibel worden erkent door de International Telecommunication Union.

Specifiek voor coaxiale lijnen geldt:  $L = \frac{\mu_0 \mu_r}{2\pi} \ln \frac{D}{d}$ ,  $C = \frac{2\pi \varepsilon_0 \varepsilon_r}{\ln \frac{D}{d}}$ . Verder kan admittantie geschreven worden als  $Y = G + jB (= Z_0^{-1})$  met de geleidbaarheid  $G$  en de susceptantie  $B$ . De susceptantie kan geschreven worden als  $B = \omega C = \frac{4\pi^2 f \varepsilon_0 \varepsilon_r}{\ln \frac{D}{d}}$ . Vervolgens wordt er gekeken naar het reële deel van de admittantie.  $G = B * \tan(\delta) = \frac{4\pi^2 f \varepsilon_0 \varepsilon_r}{\ln \frac{D}{d}} * \tan(\delta)$ . Als er nu terug naar de algemene formule wordt gekeken, dan wordt de volgende formule voor de diëlektrische demping verkregen:

$$\alpha_D = \frac{4\pi^2 f \varepsilon_0 \varepsilon_r}{\ln \frac{D}{d}} \frac{1}{2\pi} \sqrt{\frac{\mu_0 \mu_r}{\varepsilon_0 \varepsilon_r}} \ln \left( \frac{D}{d} \right) \frac{1}{2} [Np/m] \quad (3.11)$$

De verliestangens  $\tan(\delta)$  is een parameter voor diëlektrische materialen die beschrijft in welke mate er een elektromagnetische veld wordt uitgezonden. De verliestangens kan worden voorgesteld met volgend formule:  $\tan(\delta) = \frac{\varepsilon''}{\varepsilon'}$ . Om dit beter te begrijpen moet er gekeken naar de permitiviteit.

De permitiviteit is eigenlijk een complex getal:  $\varepsilon = \varepsilon' + j\varepsilon''$  waarbij  $\varepsilon'' = \frac{\sigma}{\omega}$ <sup>11</sup> en  $\varepsilon' = \varepsilon_r \varepsilon_0$ .

De permitiviteit is een maat voor hoeveel weerstand er ontstaat wanneer er een elektrisch veld wordt gegenereerd in vacuüm. In andere woorden: de permitiviteit is de maat voor hoe een elektrisch veld een diëlektrische medium beïnvloed en omgekeerd.

Bij het ontwerp van een transmissielijn zijn de gebruikte materialen heel belangrijk. Wanneer bijvoorbeeld de isolatie tussen de mantel en binnenkern slecht gekozen worden, is er geleiding tussen de twee. Meestal zijn deze verliezen te verwaarlozen tenzij er halfgeleiders gebruikt worden als isolatiemateriaal.

Algemeen mag er gezegd worden dat wanneer de diëlektrische weerstand enkele Mohm/meter bedraagt, de bijdrage tot de demping mag worden verwaarloosd. Daarnaast is deze demping onafhankelijk van de frequentie en van de vorm van het medium. Indien de demping toch van toepassing blijkt te zijn kan volgende formule gebruikt worden om de demping voor coaxiale lijnen in rekening te brengen.

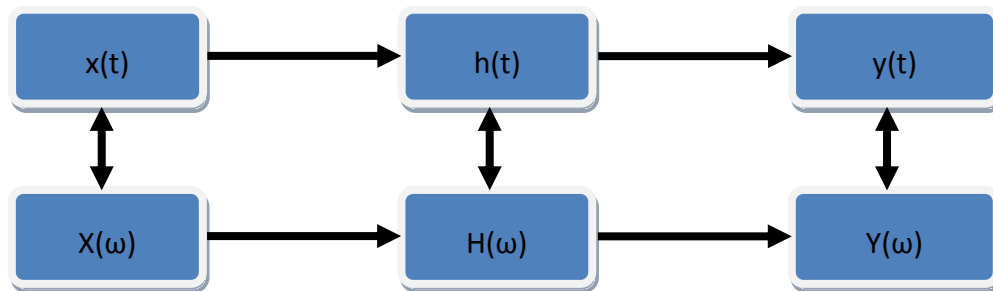
$$G' = \frac{\text{geleidbaarheid}}{\text{lengte}} = \frac{2\pi\sigma}{\ln \frac{D}{d}} \left[ \frac{\text{mohm}}{m} \right] \quad (3.12)$$

$$\alpha_G = \frac{G' Z_0}{2} = \frac{2\pi\sigma}{\ln \frac{D}{d}} \frac{1}{2\pi} \sqrt{\frac{\mu_0 \mu_r}{\varepsilon_0 \varepsilon_r}} \ln \left( \frac{D}{d} \right) \frac{1}{2} \left[ \frac{Np}{m} \right] \quad (3.13)$$

<sup>11</sup>  $\sigma$ : de conductiviteit van het materiaal.

Ten slotte is er nog het verlies als het gevolg van straling. Door de structuur van een coaxiale kabel wordt deze factor tot een minimum beperkt wordt en draagt weinig bij tot de demping. Daarom wordt deze demping niet verder besproken.

Het uiteindelijke doel van deze afleiding is om het gedrag van de coaxiale verbinding te kunnen voorspellen. Hiervoor wordt er gebruik gemaakt van een transferfunctie  $H(\omega)$ . Een transferfunctie is een functie in het frequentiedomein dat een verband weergeeft tussen input en output. Door gebruik te maken van de Fouriertransformatie kan een tijdsfunctie getransformeerd worden naar een frequentiefunctie en omgekeerd. Volgend schema maakt duidelijk welke transformaties er gedaan kunnen worden in het tijdsdomein en het frequentiedomein.



*Figuur 18: Transformatieschema*

Indien de differentiaalvergelijking 3.1 opgelost wordt, wordt volgende formule verkregen:

$$V(x) = Ve^{-\gamma z} = Ve^{-(\alpha + j\beta)z} = Ve^{-\alpha z} e^{-j\beta z} = |H(\omega)| e^{j\arg(H(\omega))} \quad (3.14)$$

De transferfunctie bestaat uit twee delen: de amplitudekarakteristiek  $|H(\omega)|$  en de fasekarakteristiek  $\arg(H(\omega))$ . De amplitudekarakteristiek kan ingevuld worden op twee manieren. Enerzijds kan de formule 3.5 gebruikt worden, anderzijds is het ook mogelijk om te werken met formule 3.14. Voor de fasekarakteristiek wordt formule 3.5 gebruikt.

### 3.1 Fasesnelheid en groepssnelheid

De fasesnelheid van een signaal is de maat voor de manier waarop de fase zich over het medium voortplant. In andere woorden, het is de snelheid waarmee de fase van het signaal verandert bij een bepaalde frequentie. De fysische interpretatie van onderstaande formule is: de fasesnelheid is gelijk aan het aantal golflengtes die passeren voorbij een vast punt per tijdseenheid.[1]

$$v_p = \frac{\lambda}{T} \left[ \frac{m}{s} \right]$$

De fasesnelheid kan ook bekeken worden vertrekkende van de pulsatie  $\omega$ . Deze geeft weer hoeveel radialen er per tijdseenheid voorbijgaan (op een bepaalde plaats). Het golfgetal  $k$  is een golfeigenschap die weergeeft hoeveel golflengtes er per lengte-eenheid voorbijkomen. Stel bijvoorbeeld dat  $k=2$ , dan legt de golf 2 golflengtes af per meter.

Indien de pulsatie  $\omega$  gedeeld wordt door het golfgetal  $k$ , wordt de fasesnelheid verkregen. Fysisch kan dit geïnterpreteerd worden als de faseverandering per seconde op een bepaald punt. Onderstaande formules gelden voor één bepaalde frequentie.

$$\begin{aligned}\omega &= 2\pi * f \left[ \frac{rad}{s} \right] \\ k &= \frac{2\pi}{\lambda} \left[ \frac{rad}{m} \right] \\ v_p &= \frac{\omega}{k} = \frac{2\pi f}{k} = \frac{2\pi}{Tk} \left[ \frac{m}{s} \right]\end{aligned}$$

Een signaal is nooit een zuivere sinusgolf maar een verzameling van sinusgolven die samen een spectrum vormen. Dankzij de Fourierserie analyse kunnen de verschillende frequentiecomponenten berekend worden. Zo bestaat een rechthoekspuls uit een oneindig aantal sinusfuncties met elk zijn eigen pulsatie. Het is pas in deze context dat er gesproken kan worden over groepssnelheid. De definitie van groepssnelheid  $v_g = \frac{d\omega}{dk}$ . Wanneer dispersie plaatsvindt op het medium dan zal  $\frac{d\omega}{dk}$  afhankelijk zijn van de frequentie.

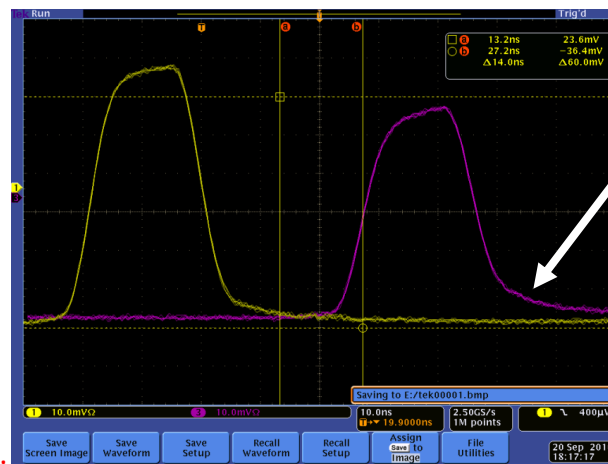
## 3.2 Dispersie

Dispersie is een fenomeen dat ontstaat wanneer de snelheid van de golf afhankelijk is van de frequentie. Er bestaan twee klassen dispersie: chromatische dispersie en multipath dispersie.

Een elektrisch signaal op een transmissielijn is een verzameling van sinusoïdale golven. Elke sinusgolf heeft zijn eigen frequentiecomponent. In deze context wordt de groepsnelheid van de golf belangrijk.

Pas wanneer er sprake is van een verzameling van frequentiecomponenten kan er gesproken worden over dispersie. Dispersie ontstaat doordat elke frequentiecomponent op het medium zijn eigen fasesnelheid heeft. Er zullen dus frequentiecomponenten vroeger aankomen dan andere. Hierdoor wordt het signaal uitgerekt in de tijd. Daarnaast is dispersie ook een gevolg van de beperkte bandbreedte van het medium. Hierdoor worden hogere frequenties meer gedempt tijdens het propageren.

Uit onderstaande figuur kan duidelijk afgeleid worden dat dispersie een fenomeen is waarbij verschillende frequentiecomponenten tegen verschillende snelheden door het medium propageren. Voor de meting werden er twee coaxiale kabels (DRAKA) van 9m aan elkaar gekoppeld. In het midden werd met een T-stuk afgetakt. Het signaal ter hoogte van het aftakpunt wordt voorgesteld door de gele plot. Het signaal op het einde van de kabel stelt de paarse plot voor.



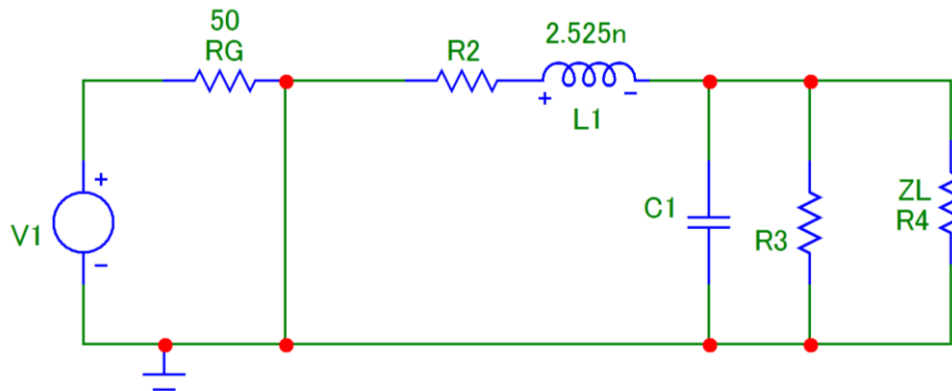
Figuur 19: Dispersie op kabel 9m (geel) – 18m (paars)

Uit bovenstaande figuur is duidelijk dat dispersie meer tot uiting komt naarmate de lengte van de kabel toeneemt. De dispersieve start wordt groter met de lengte. Dit is logisch omdat de looptijdverschillen tussen de frequentiecomponenten groter worden met de lengte. Bijvoorbeeld: een signaal heeft twee frequentiecomponenten  $f_1$  en  $f_2$ .  $f_1$  heeft een fasesnelheid van 10 m/s,  $f_2$  heeft een snelheid van 12 m/s. Als de lengte van het medium 10m bedraagt, heeft  $f_1$  1s nodig om door het medium te propageren.

Daarentegen heeft  $f_2$  0,83s nodig. Indien de lengte vergroot wordt tot 100m.  $t_{f1}=10s$ ,  $t_{f2}=8,33s$ . Het relatieve verschil blijft gelijk maar het absolute verschil wordt groter.

In het algemeen: wanneer de fasesnelheid op het medium afhankelijk is van de frequentie, dan verschilt de groepssnelheid van de fasesnelheid. Dan is de groepssnelheid ook afhankelijk van de frequentie.

### 3.3 Reflecties en afsluiting van de lijn



Figuur 20: Reflectie

Wanneer een signaal over een transmissielijn propageert, kan op elk moment zowel de stroom als de spanning worden bepaald. Het verband tussen deze stroom en spanning is de karakteristieke impedantie  $Z_0 = \frac{V_i(x,t)}{I_i(x,t)}$ .

Indien de spanning en de stroom niet volledig worden opgenomen in de lastimpedantie  $Z_L$ , dan worden de totale spanning en stroom op het medium gelijk aan:

$$\begin{aligned} V_{totaal} &= V_i(x,t) + V_r(x,t) \\ I_{totaal} &= i_i(x,t) + i_r(x,t) \end{aligned}$$

Bovenstaande vergelijking kunnen met elkaar gecombineerd worden. Na combinatie wordt er een uitdrukking verkregen dat de reflectiecoëfficiënt weergeeft. Voor meer informatie wordt er verwezen naar appendix III.

$$\rho = \frac{Z_L - Z_0}{Z_L + Z_0}$$

Als  $Z_L = Z_0$ , dan is de reflectiecoëfficiënt gelijk aan 0 en is er geen reflectie.

Als  $Z_L = \infty$ , dan zal de reflectiecoëfficiënt gelijk zijn aan 1 en is er totale reflectie met tegengestelde amplitude.

Als  $Z_L = 0$ , dan is de reflectiecoëfficiënt gelijk aan -1 en is er totale reflectie.

## 4 Analyseren van de coaxiale transmissielijn

### 4.1 DRAKA - coaxiale kabel



Draka is de naam van een internationale kabelfabrikant. Het is verantwoordelijk voor de ontwikkeling en productie van de coaxiale kabel die gebruikt wordt om de modules van CASTOR te verbinden met de QIE-kaarten.

De karakteristieken van de kabel zijn volgende: [9]

Binnenkern	0,48mm
Mantel	1,50mm
Karakteristieke impedantie	50±2 Ohm
Snelheidsratio	0,66 %
Capacitantie	101pF/m
DC weerstand binnenkern	126 mOhm/m
DC weerstand mantel	40 mOhm/m

*Tabel 3: Kabelspecificatie*

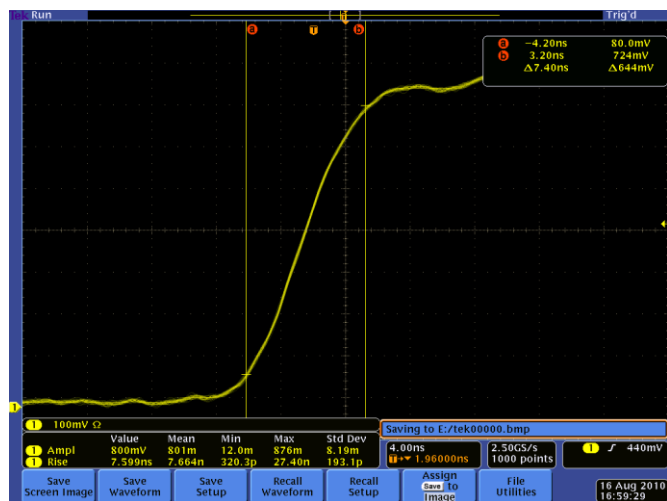
### 4.2 Rise time - Bandbreedte

Aan de hand van de rise time van een puls kan de bandbreedte van de kabel gemeten worden. De rise time is de tijd dat het signaal nodig heeft om van 10% van de amplitude op te klimmen naar 90% van de amplitude.

Het verband tussen de rise time en de bandbreedte wordt weergegeven aan de hand van volgende formule:  $t_{rise} = \frac{0,34}{BW} \Leftrightarrow BW = \frac{0,34}{t_{rise}}$ . Hierbij wordt aangenomen dat de kabel kan worden benaderd als een RC filter.

Tijdens volgende metingen werd gebruik gemaakt van de Keithley 3390 als functiegenerator. Deze functiegenerator heeft een bandbreedte van 50MHz en een stijgtijd van 5ns. De oscilloscoop die gebruikt werd voor het opnemen van de metingen is: Tektronix DPO 4054. Hij heeft een bandbreedte van 500MHz en een stijgtijd van 700ps.

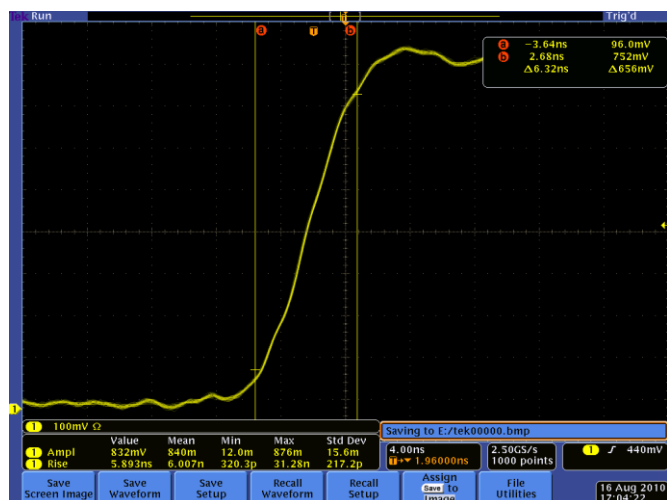
## Lange kabel van 64ns<sup>12</sup>



Figuur 21: Rise time bepaling kabel 64ns

De kabel van 64ns heeft een rise time van 7,66 ns. Dit resulteert in een bandbreedte van 44,3 MHz.

## Korte kabel van 8 ns



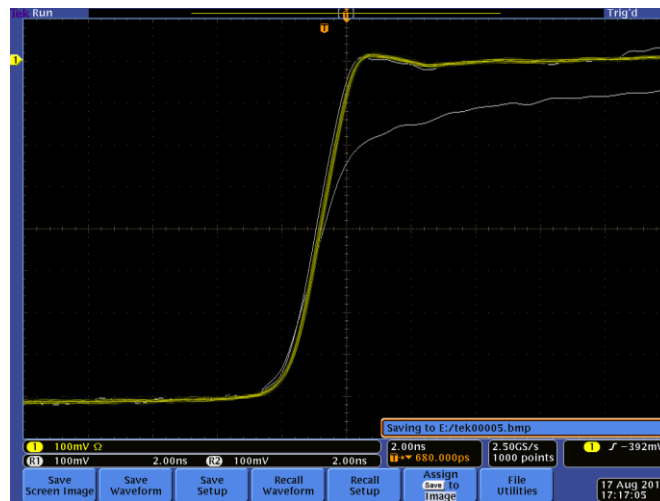
Figuur 22: Rise time bepaling kabel 8ns

Deze kabel heeft een rise time van 6 ns. Dit resulteert in een bandbreedte van 56,6 MHz.

In volgende opstelling werden drie lengten van kabels gebruikt: 8ns, 16ns en 64ns. Het signaal van de langste kabel is het minst steil en heeft dus de grootste rise time. Daarnaast

<sup>12</sup> Dit wil zeggen dat het signaal 64 ns nodig heeft om door de kabel te propageren. Indien de lengte van de kabel gekend moet zijn, kan de velocity ratio van de kabel specificatie geraadpleegd worden. Vervolgens kan de lengte berekend worden via  $v = \frac{dx}{dt}$ .

heeft de 64ns-kabel ook al last van een zekere hoeveelheid demping.  
Tussen het signaal van de kabel van 1 of 3 meter is weinig verschil. Het verschil in rise time is miniem.

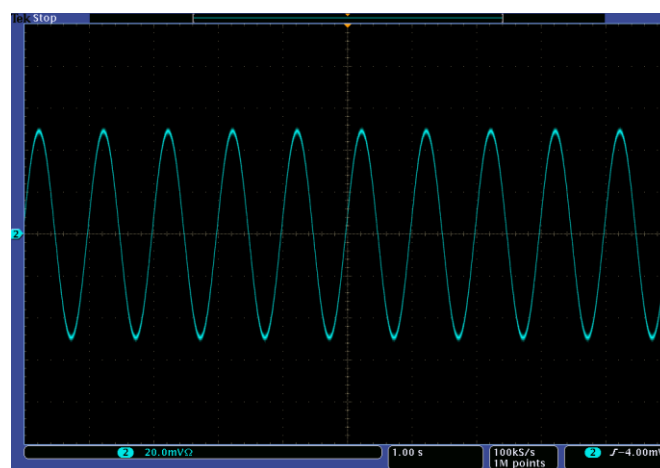


Figuur 23: Rise time vergelijking 8ns - 16ns - 64 ns

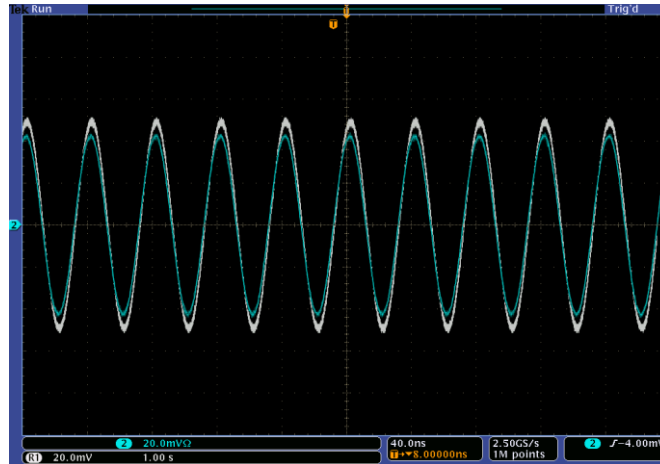
### 4.3 De frequentieafhankelijkheid van de demping

Meting 1: Hier is er een sinusoidaal signaal met amplitude 100mV en een frequentie van 1Hz aangelegd op een coaxiale kabel van 64ns. Vervolgens wordt de frequentie verhoogd tot 25 MHz. Op volgende figuren kan er gezien worden dat de amplitude van het signaal afgenomen is tot 80mV.

Conclusie: Indien de frequentie van 1 Hz naar 25MHz toeneemt, daalt de amplitude van 100mV naar 80mV. Hoe groter de frequentie, hoe groter de dempingconstante.



Figuur 24: Sinus 1Hz en Amplitude 100mV



Figuur 25: Amplitude sinus 25MHz

In volgende tabellen worden de meetresultaten geïllustreerd die de frequentieafhankelijkheid van de demping in kaart brengen. De metingen werden opgenomen met een oscilloscoop met een bandbreedte van 500MHz. Om ervoor te zorgen dat de metingen niet beïnvloed worden door de bandbreedtebeperking van de oscilloscoop is de maximale frequentie waarvoor er een meting is gebeurd gelijk aan 300MHz. De signaalgenerator genereert een sinusoïdaal signaal met een amplitude van 325mV.

Tabel 4 bevat meetresultaten van drie testkabels met telkens een andere lengte. Tabel 5 bevat de resultaten van de metingen op de coaxiale kabel van DRAKA. De kabel gebruikt tijdens de meting heeft een lengte van 9m. Indien dit wordt omgerekend naar tijd, geeft dit  $44,8\text{ns}^{13}$ . Nu kan er een vergelijking gemaakt worden tussen de verschillende kabels.

Frequentie	Amplitude 8 ns [mV]	Amplitude 16 ns [mV]	Amplitude 64 ns [mV]
1 Hz	324	321	320
1 kHz	324	322	319
1 MHz	315	311	309
10 MHz	308	298	278
50 MHz	300	287	236
100 MHz	288	272	199
150 MHz	278	257	183
200 MHz	269	249	162
250 MHz	260	236	147
300 MHz	252	226	134

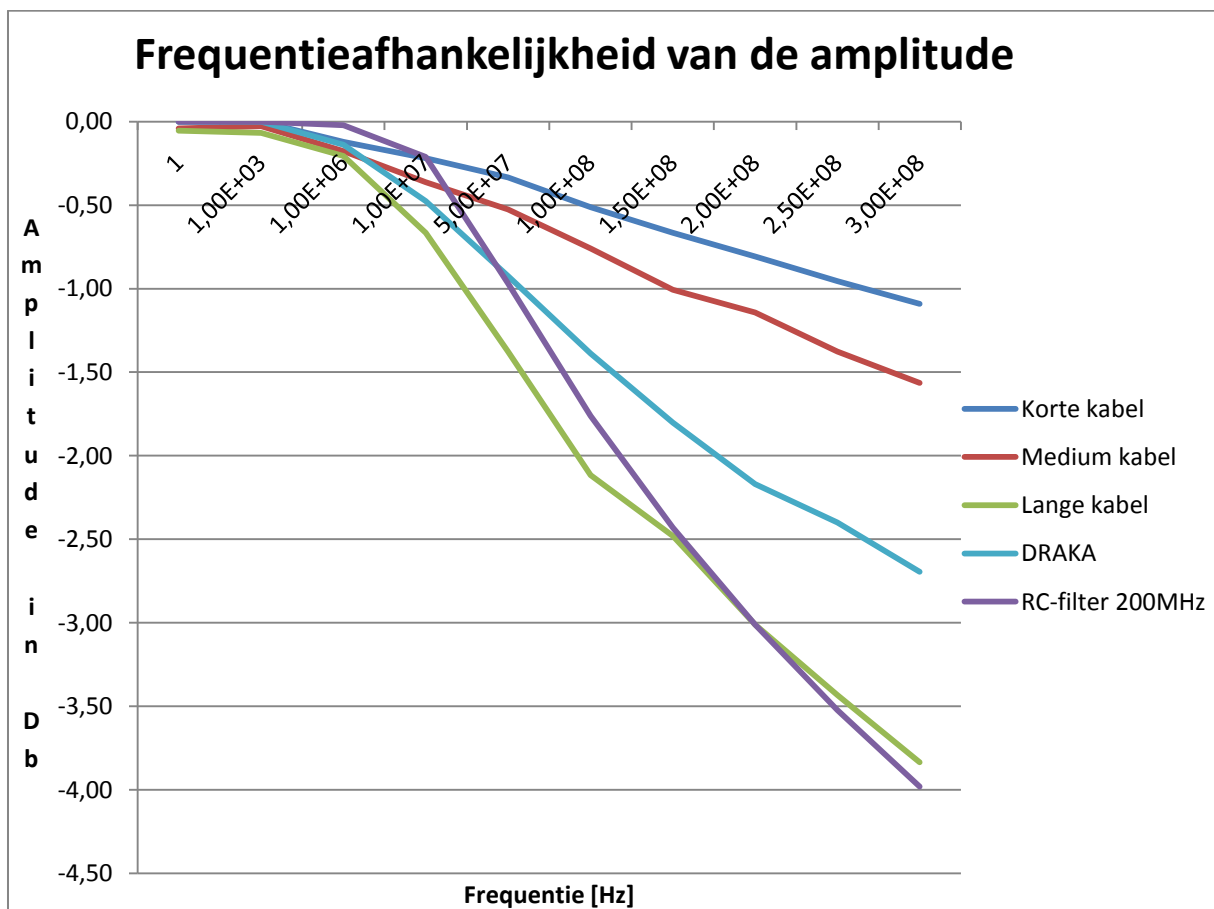
Tabel 4: Amplitude i.f.v. frequentie

<sup>13</sup>  $\frac{9}{3E8 * 0,67} = 4,48E^{-8} = 44,8E^{-9} = 44,8\text{ns}$

Frequentie	Amplitude [mV] 44,8ns kabel
1 Hz	318
1 kHz	318
1 MHz	308
10 MHz	285
50 MHz	257
100 MHz	231
150 MHz	210
200 MHz	193
250 MHz	183
300 MHz	171

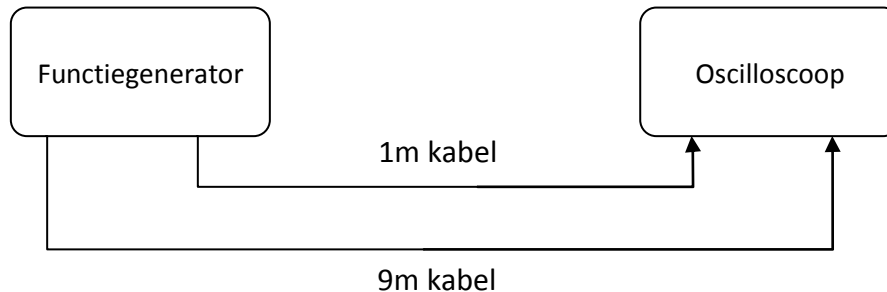
Tabel 5: Amplitude i.f.v. frequentie DRAKA

Uit vorige tabellen blijkt nogmaals dat wanneer de frequentie toeneemt, de demping toeneemt. Verder kan vastgesteld worden dat de lengte van de kabel ook invloed heeft op de demping. In volgende grafiek is de frequentie en amplitude geplot in dB.



#### 4.4 De frequentieafhankelijkheid van de fase

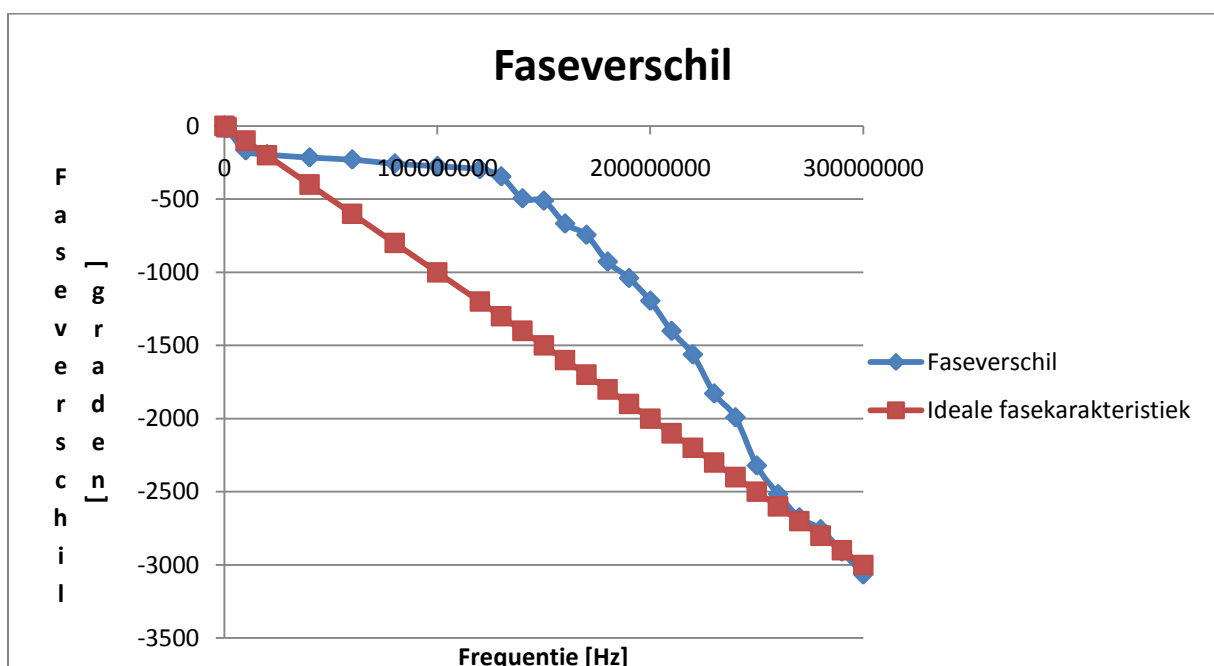
Voor het meten van een faseverschil, is onderstaande opstelling nodig. Een korte kabel zal gebruikt worden als referentie. De fase van het signaal op de andere kabel zal vergeleken worden met de referentiekabel.



Figuur 26: Opstelling fasemeting

Over de kabel van 1m loopt het originele signaal. De faseverschuiving ten gevolge van de frequentie wordt op deze kabel verwaarloosd. De twee kabels worden elk op een andere kanaal van de oscilloscoop aangesloten. De oscilloscoop berekent het faseverschil tussen beide signalen. De metingen moeten zeer aandachtig gebeuren omdat de oscilloscoop geen rekening houdt met het totale faseverschil. De oscilloscoop bepaalt het faseverschil binnen één periode. Elke meting moet nauwkeurig geïnterpreteerd worden en indien nodig moet er  $180^\circ$  of  $\pi$  radialen bij het faseverschil opgeteld worden. De tabel met de geïnterpreteerde waarden is terug te vinden in appendix IV.

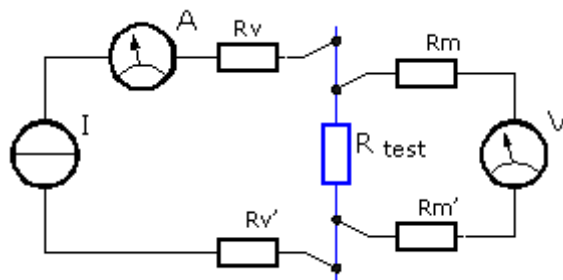
Op onderstaande grafiek wordt het verband weergegeven tussen de frequentie en het faseverschil. De ideale fasekarakteristiek heeft een lineair verloop. Een lineair faseverloop verzekert dat er geen delen van het signalen in de tijd verschoven zijn.



Besluit: uit bovenstaande grafiek kan afgeleid worden dat het faseverschil toeneemt naargelang de frequentie toeneemt maar niet de ideale karakteristiek volgt. Er wordt ook vastgesteld dat de gemeten fase nait op de ideale fasekarakteristiek.

#### 4.5 Vierpuntsmeting voor het bepalen van de weerstand van de kabel

Op Figuur 27: Opstelling vierpuntsmeting wordt de meetopstelling grafisch voorgesteld. De meting gebeurt aan de hand van een gescheiden spannings- en stroommeting. Er wordt gebruik gemaakt van een stroombron om ervoor te zorgen dat de meetsnoeren en contacten geen invloed meer hebben op de stroom.



*Figuur 27: Opstelling vierpuntsmeting*

De vierpuntsmeting werd zowel op de binnenkern als op de mantel van de coaxiale kabel met lengte 9m uitgevoerd. De binnenkern heeft een weerstand van 1,167 Ohm. De weerstand van de mantel is gelijk aan 0,284 Ohm.

Indien deze waarden vergeleken worden met de opgegeven waarden door de fabrikant DRAKA, wordt er vastgesteld dat de gemeten waarden voor de binnenkern ongeveer gelijk is aan de vooropgestelde waarden. De gemeten waarden voor de mantel wijkt  $\pm 25\%$  af.

	DRAKA	gemeten
DC weerstand binnenkern	126 mOhm/m	129 mOhm/m
DC weerstand mantel	40 mOhm/m	31,6 mOhm/m

*Tabel 6: Resultaten vierpuntsmeting*

## 5 Opstellen van het mathematische model van de kabel

### 5.1 Gebruikte software: Matlab



Matlab is een technische softwareomgeving uitgegeven door The Mathworks. Het softwarepakket wordt vaak gebruikt zowel in de industrie als in de academische wereld. Matlab kent vele toepassingen: het berekenen van functies, werken met matrices, gebruik van fourier- en laplacetransformaties. Indien er veel en complexe berekeningen gedaan moeten worden is het interessant om een matlabscript te schrijven. Dit script moet geschreven worden in M-code.[11]

### 5.2 Basic fitting op gemeten amplitude- en fasekarakteristiek

De functionaliteit “Basic Fitting” in Matlab maakt het mogelijk om van een aantal opgegeven punten een functievoorschrift op te stellen. Hierbij kan er gekozen worden voor verschillende benaderingen zoals bijvoorbeeld: lineair, kwadratisch,  $n^{\text{de}}$  orde polynomen.

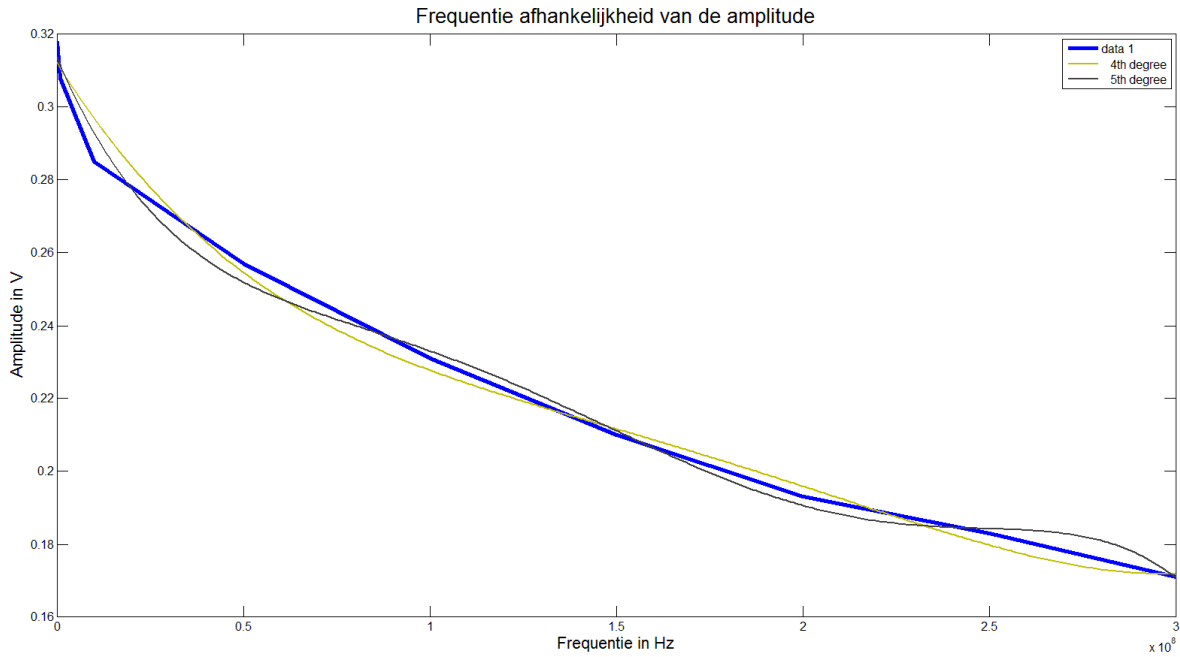
De gegevens van Tabel 5: Amplitude i.f.v. frequentie DRAKA worden gebruikt om een grafiek op te stellen in Matlab. Dit gebeurt aan de hand van volgende commando's:

```
freq=[1 1E3 1E6 1E7 5E7 1E8 1.5E8 200E6 250E6 300E6];  
amp=[318 318 308 285 257 231 210 193 183 171]/1000;  
figure(1);  
plot(freq,amp,'LineWidth',4);
```

Vervolgens wordt er gebruik gemaakt van de toolbox ‘Basic Fitting’. De toolbox kan gevonden worden onder Tools. Basic Fitting maakt het mogelijk om verschillende benadering te berekenen op een aantal opgegeven punten.

Voor de benadering van de gemeten amplitudekarakteristiek en fasekarakteristiek werden  $4^{\text{de}}$  en  $5^{\text{de}}$  orde polynomen berekend. De polynomen van de amplitudekarakteristiek worden geplotted in onderstaande figuur. De dikkere blauwe lijn stelt de oorspronkelijke data voor. De geelgroene lijn representeert de  $4^{\text{de}}$  orde polynoom. De  $5^{\text{de}}$  orde polynoom wordt voorgesteld door de zwarte lijn.

Een polynoom benadert de oorspronkelijk data. De orde van het polynoom zal bepalen hoe groot de fout is ten opzichte van de oorspronkelijke data. Hierbij geldt: hoe hoger de orde van de polynoom, hoe kleiner de fout tussen de polynoom en de data.



Figuur 28: Curve-fit van amplitudekarakteristiek

Afhankelijk van de manier waarop de curve-fit plaatsvindt, wordt andere informatie verkregen. In onderstaande tabel staat het voorschrift van de 4<sup>de</sup> en 5<sup>de</sup> orde polynoom met telkens de bijhorende coëfficiënten.

4 <sup>de</sup> orde polynoom	5 <sup>de</sup> orde polynoom
$y = p1 \cdot x^4 + p2 \cdot x^3 + p3 \cdot x^2 + p4 \cdot x + p5$	$y = p1 \cdot x^5 + p2 \cdot x^4 + p3 \cdot x^3 + p4 \cdot x^2 + p5 \cdot x + p6$
Coefficients: p1 = 6.3588e-035 p2 = -4.5615e-026 p3 = 1.1862e-017 p4 = -1.6381e-009 p5 = 0.31226	Coefficients: p1 = -1.1223e-042 p2 = 8.9671e-034 p3 = -2.6009e-025 p4 = 3.4004e-017 p5 = -2.3908e-009 p6 = 0.31378

Hetzelfde werd gedaan worden voor de fasekarakteristiek. Hierbij worden volgende polynomen verkregen.

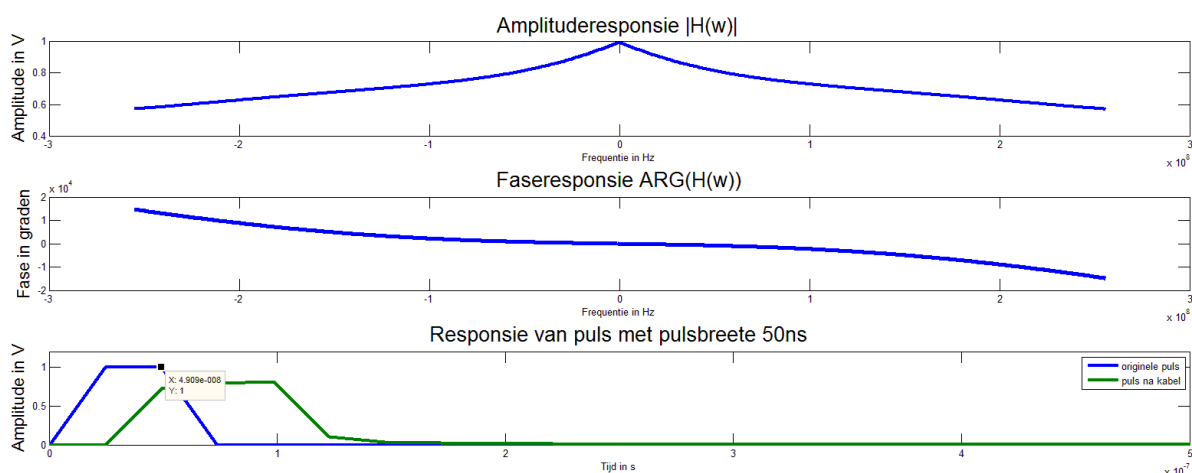
4 <sup>de</sup> orde polynoom	5 <sup>de</sup> orde polynoom
$y = p1 \cdot x^4 + p2 \cdot x^3 + p3 \cdot x^2 + p4 \cdot x + p5$	$y = p1 \cdot x^5 + p2 \cdot x^4 + p3 \cdot x^3 + p4 \cdot x^2 + p5 \cdot x + p6$

<p>Coefficients:</p> <p>p1 = 1.8507e-030</p> <p>p2 = -1.1621e-021</p> <p>p3 = 1.8978e-013</p> <p>p4 = -1.2668e-005</p> <p>p5 = -3.7002</p>	<p>Coefficients:</p> <p>p1 = 1.0477e-038</p> <p>p2 = -5.9723e-030</p> <p>p3 = 8.8312e-022</p> <p>p4 = -2.814e-014</p> <p>p5 = -4.7156e-006</p> <p>p6 = -33.188</p>
--	--

Indien bovenstaande karakteristieken gebruikt worden om de coaxiale transmissielijn te simuleren, worden onderstaande grafieken verkregen. De amplitude- en fasekarakteristiek bekomen via de basis fitting wordt vertaald naar respectievelijk een amplituderesponsie  $|H(w)|$  en een faseresponsie  $ARG(H(w))$ .

De amplituderesponsie dient een even functie te zijn, d.w.z. dat de functie symmetrisch moet zijn met als symmetrieas de y-as. Het faseverloop moet een oneven functie zijn. Een oneven functie is een functie die symmetrisch is ten opzichte van de oorsprong.

Deze responsie in het frequentiedomein kan getransformeerd worden naar het tijdsdomein via een inverse fouriertransformatie. Vervolgens wordt een puls met de impulsresponsie geconvolueerd. Het verloop van deze puls na convolutie wordt voorgesteld in de onderste plot.

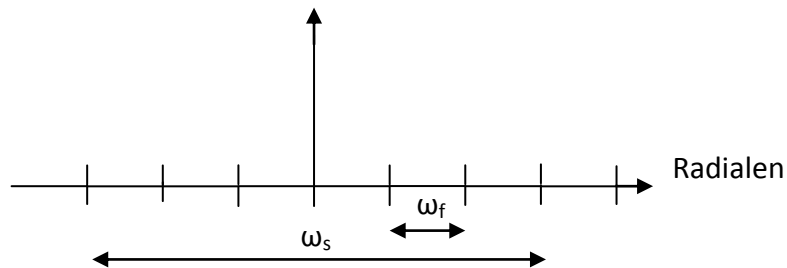


*Figuur 29: Responsie via basic fitting*

Bij bovenstaande tijdsresponsie moet er opgemerkt worden dat er een puls van met een breedte van 50ns gebruikt werd. De reden hiervoor is dat de amplitude- en fasekarakteristiek zijn opgemeten tot 300MHz. De gebruikte oscilloscoop heeft een bandbreedte van 500MHz. Er werd besloten om op 300MHz met de meting te stoppen om

geen bandbreedte gerelateerde invloeden te krijgen. De relatie tussen het frequentiebereik van de meting en de tijdsresolutie wordt duidelijk aan de hand van Figuur 30: Het frequentiedomein en Figuur 31: Het tijdsdomein.

Op onderstaande figuren wordt aangenomen dat een signaal een beperkte bandbreedte  $\omega_B$  heeft en beperkt is in de tijd. Op de eerste figuur wordt de frequentie-as weergegeven.  $\omega_s$  stelt de samplefrequentie voor. Volgens het Nyquist criterium moet  $\omega_s$  groter of gelijk zijn aan  $2 \times \omega_B$ . De frequentieresolutie  $\omega_f$  bepaalt hoe groot de frequentiestappen zijn.  $\omega_s$  en  $\omega_f$  bepalen samen het aantal samples  $N (= \frac{\omega_s}{\omega_f})$ . [8]

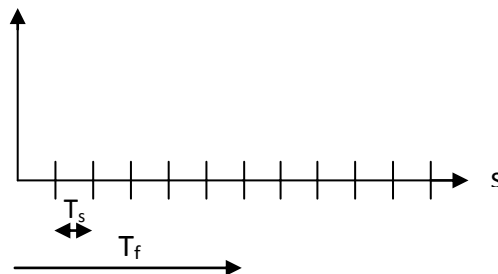


*Figuur 30: Het frequentiedomein*

Aan de hand van onderstaande formules, wordt de relatie aangetoond tussen het frequentiedomein en het tijdsdomein. De samplefrequentie bepaalt de sampleperiode  $T_s$  in het tijdsdomein. De frequentieresolutie bepaalt de periode  $T_f$ .

$$T_s \omega_s = 2\pi \Leftrightarrow T_s = \frac{2\pi}{\omega_s}$$

$$T_f \omega_f = 2\pi \Leftrightarrow T_f = \frac{2\pi}{\omega_f}$$



*Figuur 31: Het tijdsdomein*

De metingen uit paragraaf 4.3 en 4.4 kunnen beschouwd worden als een frequentiesampling. De maximale frequentie waarbij een meting is gebeurd, is gelijk aan 300MHz. Dit heeft tot gevolg dat de tijdsresolutie gelijk is aan  $10\text{ns}^{14}$ . De resultaten bekomen aan de hand van de basic fitting kunnen dus niet vergeleken worden met de PMT-pulsen vanwege een te kleine tijdsresolutie. In appendix V wordt meer uitleg gegeven over de techniek zeropadding.

Hierbij moet opgemerkt worden dat de bekomen polynomen maar betrouwbaar zijn tot 300MHz. Een eventuele oplossing hiervoor kan zijn om handmatig punten toe te voegen via extrapolatie. Om een tijdsresolutie te verkrijgen van 1ns, moet de frequentie gesampled worden tot 3,14GHz ( $\frac{\omega_s}{2} = \frac{2\pi}{1\text{E}-9} * 0,5$ ). Dit betekent dat 92,75%<sup>15</sup> van de frequentiepunten afkomstig zijn van extrapolatie i.p.v. de meting. Daarom is extrapolatie voor het verkrijgen van de gewenste tijdsresolutie niet toepasbaar.

Toch kunnen er enkele zaken worden afgeleid uit de bekomen impulsresponsie, met name demping en dispersie. Op de onderste plot van Figuur 29: Responsie via basic fitting zijn twee pulsen te zien. De blauwe puls stelt de oorspronkelijk puls aan het begin van de kabel voor. De groene puls representeert de puls op het einde van een 9m lange coaxiale kabel van de fabrikant DRAKA. Indien de amplitude van beide pulsen met elkaar vergeleken worden, wordt vastgesteld dat de puls op het einde van de kabel in amplitude is afgenomen. Daarnaast is de puls ook langer geworden ten gevolge van dispersie. Om toch te kunnen werken met een grotere tijdsresolutie, dus een kleinere  $T_s$ , wordt de kabel in het volgende hoofdstuk benaderd aan de hand van een mathematische voorstelling.

### 5.3 Model aan de hand van een transferfunctie

#### Model:

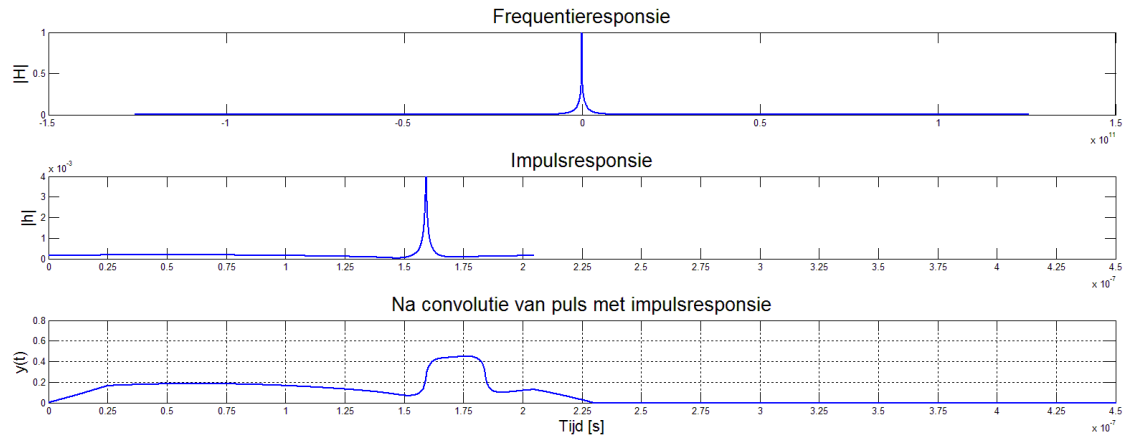
Dit model wordt afgeleid op basis van de transferfunctie voor transmissielijnen. Tijdens deze implementatie wordt rekening gehouden met het skineffect. Per frequentie wordt de weerstand van de kabel opnieuw bepaald aan de hand van de skindiepte. De zelfinductie L wordt bepaald aan de hand van volgende formule  $Z_0 = \sqrt{\frac{L}{C}}$ . C werd overgenomen uit de specificatie van de kabel en wordt als constant beschouwd.

Aan de hand van bovenstaande, wordt onderstaande transferfunctie verkregen:

$$V(w, x) = V_0 e^{-\left[ \sqrt{\frac{1}{2}[(RG - \omega^2 LC) + \sqrt{(R^2 + \omega^2 L^2)(G^2 + \omega^2 C^2)}}]} + j \sqrt{\frac{1}{2}[\sqrt{(R^2 + \omega^2 L^2)(G^2 + \omega^2 C^2)} - (RG - \omega^2 LC)]} \right] x}$$

<sup>14</sup>  $(2 * \pi) / (2 * 300\text{MHz}) = 10,4\text{ns}$

<sup>15</sup>  $((3,14\text{GHz} - 300\text{MHz}) / 6,28\text{GHz}) * 100 = 92,75\%$

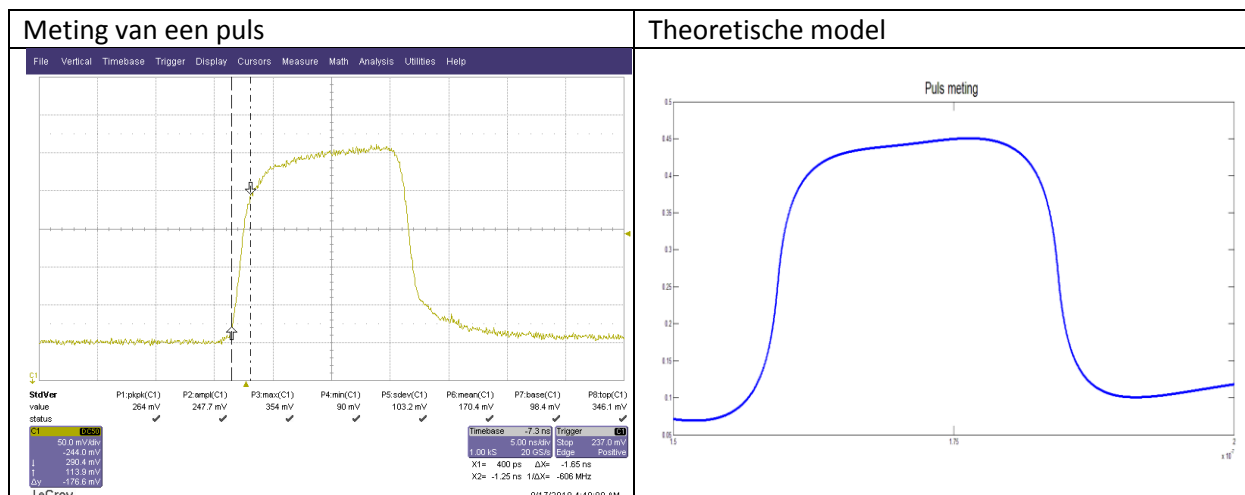


Figuur 32: Model via transferfunctie

Op de onderste plot in bovenstaande figuur wordt een puls van 50ns geconvolveerd met de impulsresponsie van de kabel. Op de plot wordt een grid getekend dat ervoor zorgt dat de verschillende time slices (25ns integratie-interval) beter van elkaar onderscheid kunnen worden. Het theoretisch model beschrijft niet de waargenomen starten van het QIE-signaal.

In onderstaande tabel wordt de meting en het theoretische model met elkaar vergeleken. Beide pulsen hebben een breedte van  $\pm 20$ ns nadat de puls over de kabel is gepropageerd.

Daarnaast moet er opgemerkt worden dat in beide gevallen de puls binnen het interval van 25ns blijft.



Tabel 7: Vergelijking meting - model

### 5.3.1 Besluit

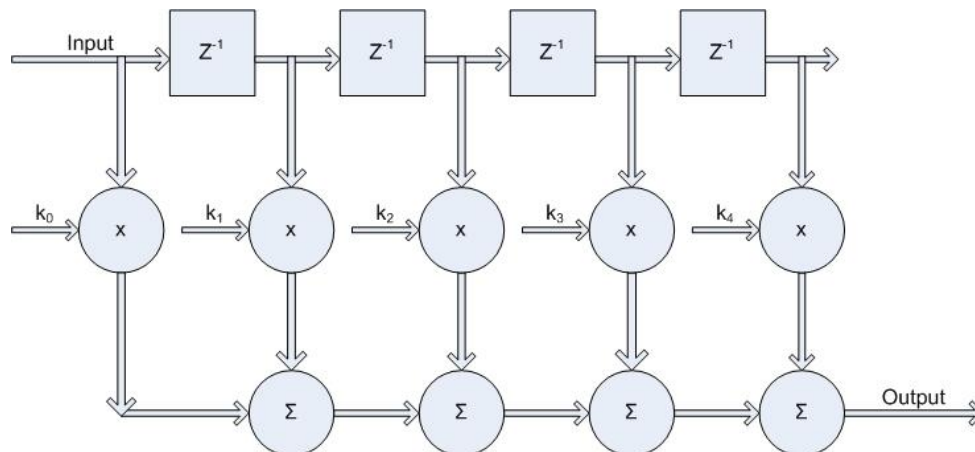
Met het theoretisch model voor transmissielijnen kunnen de waargenomen staarten van het QIE-sigitaal niet worden verklaard. Om toch een oorzaak te vinden is interessant om de QIE-kaart te onderzoeken, meerbepaald de amplitudeafhankelijke impedantie waarlangs de condensator wordt ontladen. De benodigde testopstelling om dit te onderzoeken is niet tijdig gerealiseerd. De realisatie van deze testopstelling valt buiten het kader van deze thesis.

Daar het model niet overeenkomt met de beschikbare data, kan dit niet worden gebruikt voor de ontwikkeling van een corrigerende filter.

## 6 FIR filter

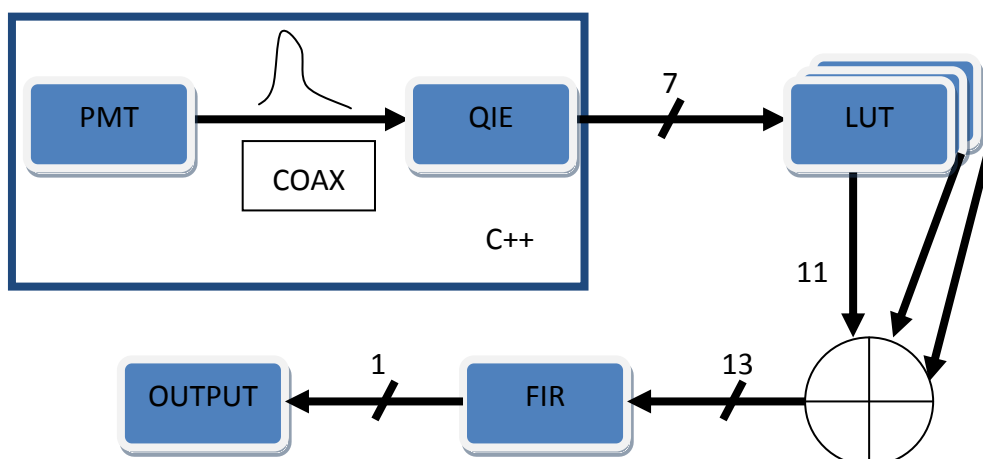
### 6.1 Wat is een FIR filter?

Een FIR<sup>16</sup> filter gebruikt van een eindige impulsresponsie om de gewenste karakteristiek te bekomen. Het filter bestaat uit een aantal geheugenelementen weergegeven door  $z^{-1}$ . Het aantal geheugenelementen is gelijk aan de orde van de filter. Hoe groter de orde, hoe meer de gewenste filterkarakteristiek bekomen wordt. Daarnaast geldt ook dat naar mate de orde van de filter toeneemt, de filter scherper wordt. Elke tak van de FIR wordt vermenigvuldigd met een coëfficiënt. De reeks van coëfficiënten kan berekend worden aan de hand van de impulsresponsie van de filter (zie paragraaf 6.2: Bepalen van filtercoëfficiënten).[8] [17]



Figuur 33: Structuur FIR filter

Bij het ontwerpen en implementeren van een filter moet er gekeken worden naar het volledige systeem. Dit kan voorgesteld worden aan de hand van volgend signaalstroomschema:



Figuur 34: Signaalstroomschema

<sup>16</sup> Finite Impulse Response

Om de coëfficiënten van het FIR filter te bepalen wordt er een simulatieomgeving gecreëerd. Het eerste gedeelte van het signaalstroomschema wordt verzorgd door een C++-programma. De andere delen worden in HDL geschreven. Niet-lineaire QIE-data wordt door het zelfgeschreven software programma verwerkt. Voor meer informatie over de dataverwerking, wordt er verwezen naar hoofdstuk 7: Software. In de software wordt de niet-lineaire QIE-data omgezet naar bitvectoren die als ingangssignalen zullen gebruikt worden tijdens de Modelsim-simulatie.

#### Het signaalpad:

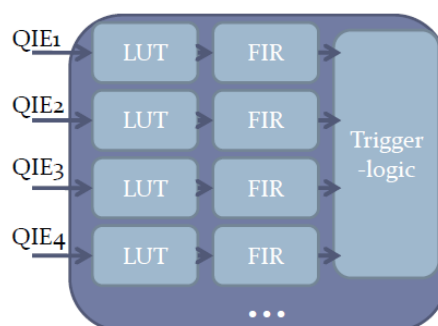
Het signaal van de PMT wordt vervormd door de coaxiale kabel. Vervolgens komt het signaal aan in de QIE-kaart. Deze digitaliseert de data (zie paragraaf 1.4: QIE). De QIE-data wordt aangeboden aan de HDL module: TriggerPath. Met behulp van een tabel, LUT<sup>17</sup>, wordt de niet-lineaire QIE-data omgezet in lineaire data.[22] Vervolgens wordt de LUT-output per drie opgeteld. Zo ontstaan er vier subtriggers: tow11, tow12, tow13 en tow14. Ten slotte worden de vier subtriggers met elkaar opgeteld en vormen het uiteindelijke triggersignaal.[7]

```
tow11 <= InLut1DOut + InLut2DOut+InLut3DOut;
tow12 <= InLut4DOut + InLut5DOut+InLut6DOut;
tow13 <= InLut7DOut + InLut8DOut+InLut9DOut;
tow14 <= InLut10DOut+InLut11DOut+InLut12DOut;
```

#### Implementatie :

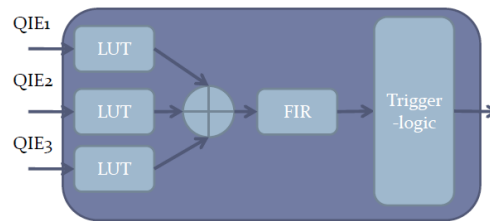
Er zijn verschillende implementaties van de filter mogelijk.

- Een eerste mogelijkheid is om per QIE-sigitaal een FIR filter te implementeren. Deze implementatie zal in hardware de meeste logische cellen innemen. Het voordeel bij deze implementatie is dat er in principe voor elk kanaal een aparte set coëfficiënten berekend kan worden en zo dus voor de beste onderdrukking van de signaalstart kan zorgen.



<sup>17</sup> LUT : Look up Table

- Daarnaast kan er ook per subtower een filter geïmplementeerd worden. Het is een afweging tussen de onderdrukking van de signaalstaart en het aantal nodige logische cellen.

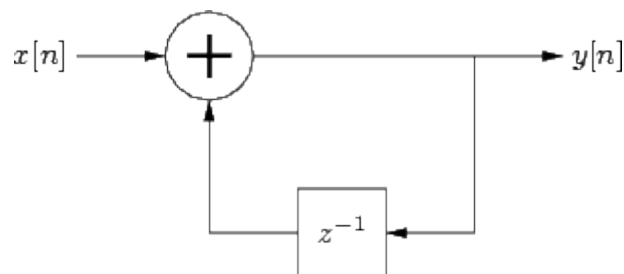


- Ten slotte kan er ook maar één filter geïmplementeerd worden per triggersignaal/sector. Daar er vier<sup>18</sup> triggers zijn, zal deze oplossing evenveel FIR filters bemoedigen als de oplossingen met de subtower en zal daarom niet verder worden uitgewerkt.

De manier van implementeren is afhankelijk zijn van verschillende factoren. Enerzijds moet er rekening gehouden worden met de hoeveelheid onderdrukking die geleverd kan worden. Anderzijds zal er gekeken moeten worden naar het aantal logische cellen die de filter zal innemen op de FPGA.

## 6.2 Alternatief: Feedback filter

Als alternatief op de FIR filter, is er het filter met feedback. Op onderstaande figuur is de structuur van dergelijk eerste orde filter te zien.



*Figuur 35: Feedback filter*

Daar de coaxiale kabel niet te beschrijven is als een eerste orde systeem zal ook het filter met feedback van een hogere orde zijn. Naar implementatie in hardware toe, is er geen of weinig verschil ten opzichte van de FIR filter.

Bij hogere orde filters van dit type, is het garanderen van de stabiliteit een moeilijke opgave omwille van afrondingen. Bovendien moet ervoor gezorgd worden dat dit type filter over een continue datastroom kan beschikken. Dit is wel het geval indien het filter wordt geïmplementeerd maar er is geen continue datastroom beschikbaar om het filter te ontwerpen. Daarom is dit soort filter niet verder in de beschouwing opgenomen.

<sup>18</sup> 2 Energietrigger: 1 per sector, 1 Noleveltrigger, 1 Muontrigger

### 6.3 Bepalen van de filtercoëfficiënten

In het tijdsdomein kunnen de coëfficiënten van de filter bepaald worden aan de hand van een stelsel lineaire vergelijkingen.

$$\begin{aligned}
 y'_0 &= k_0 v_0 \\
 y'_1 &= k_0 v_1 + k_1 v_0 \\
 y'_2 &= k_0 v_2 + k_1 v_1 + k_2 v_0 \\
 y'_3 &= k_0 v_3 + k_1 v_2 + k_2 v_1 + k_3 v_0 \\
 y'_4 &= k_0 v_4 + k_1 v_3 + k_2 v_2 + k_3 v_1 + k_4 v_0
 \end{aligned}$$

$k_n$  zijn de coëfficiënten van de filter  
 $v_n$  zijn de input samples  
 $y'_n$  zijn de gewenste waarden

Door het gebrek aan een model dat de QIE-sigitaalstarten exact beschrijft, wordt in software een gemiddelde impulsresponsie bepaald aan de hand van statistische data. Voor meer informatie over hoe de gemiddelde impulsresponsie berekend wordt, wordt er verwezen naar het hoofdstuk 7: Software. Bovenstaande vergelijkingen kunnen herschreven worden in matrixvorm. In het onderstaande voorbeeld is de gemiddelde impulsresponsie gelijk aan [1067 157 48 27 19].

$$T = \begin{bmatrix} 1067 & 0 & 0 & 0 & 0 & 1067 \\ 157 & 1067 & 0 & 0 & 0 & 0 \\ 48 & 157 & 1067 & 0 & 0 & 0 \\ 27 & 48 & 157 & 1067 & 0 & 0 \\ 19 & 27 & 48 & 157 & 1067 & 0 \end{bmatrix}$$

De laatste kolom bevat de waarde die als output van de filter verwacht worden. Er zal worden getracht om enkel de eerste waarde van de impulsresponsie, hierboven 1067, door het filter te laten passeren. De andere waarden van de impulsresponsie moeten onderdrukt worden. Om dit te bereiken zullen voor de verwachte waarde nullen ingevuld worden. Nu dat de matrix is opgesteld, kan via het *rref*-commando van Matlab of soortgelijke software, opdracht gegeven worden om het stelsel vergelijkingen op te lossen. Indien bovenstaande stelsel wordt opgelost, wordt volgende matrix verkregen.

$$A = rref(T) = \begin{bmatrix} 1 & 0 & 0 & 0 & & 1 \\ 0 & 1 & 0 & 0 & -0,14714 & \\ 0 & 0 & 1 & 0 & -0,02334 & \\ 0 & 0 & 0 & 1 & -0,01525 & \\ 0 & 0 & 0 & 0 & 1 & -0,01079 \end{bmatrix}$$

De laatste kolom representeert de coëfficiënten van de filter, in dit geval een 4<sup>de</sup> order filter. Op te merken valt dat de berekende coëfficiënten relatief klein zijn, zeker indien deze geïmplementeerd moeten worden in hardware. De FIR filter is een lineaire bewerking. De coëfficiënten kunnen dus vermenigvuldigd worden met een factor. Uiteraard moet deze factor op het einde van de berekeningen gecorrigeerd worden.

## 6.4 De orde van de filter

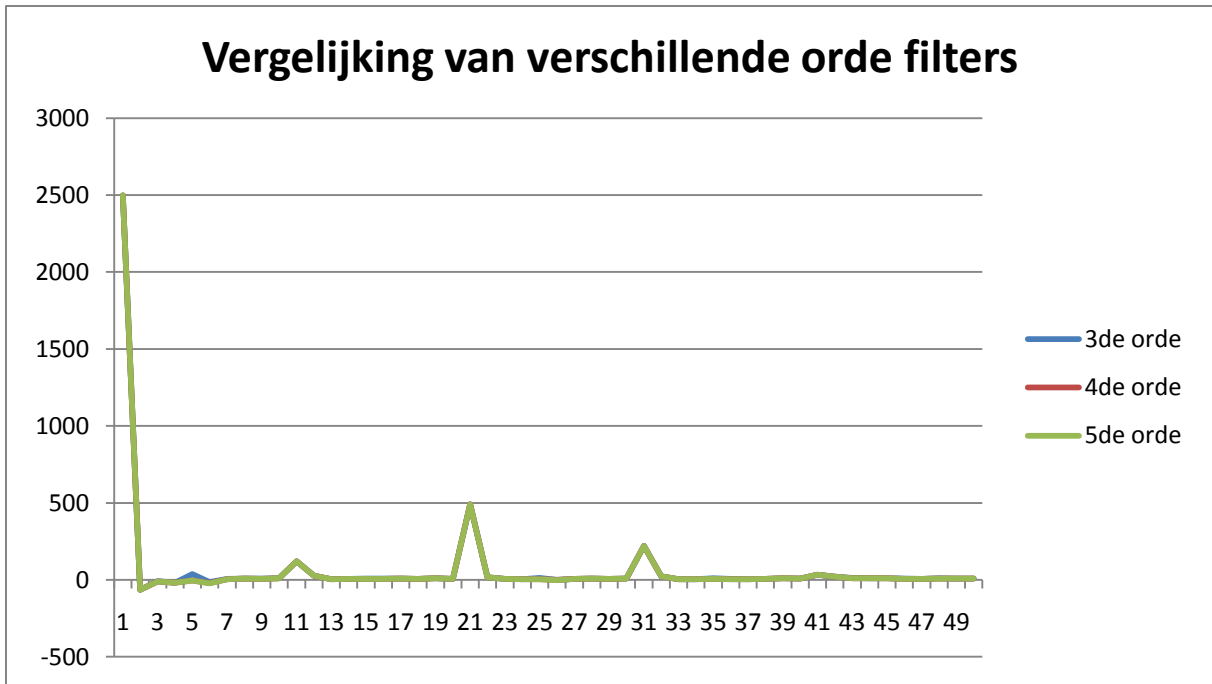
De orde van de filter wordt bepaald aan de hand van de lengte van de (gemiddelde) impulsresponsie van het ingaande systeem. Hoe hoger de orde van de filter genomen wordt, hoe meer de gewenste filterkarakteristiek bekomen wordt. Dit heeft ook een keerzijde: filters met een hogere orde hebben meer kans om een oscillatorisch gedrag te vertonen. Dit kan resulteren in een toename van het residu.

Aan de hand van onderstaande grafieken, wordt de keuze voor de orde van de filter verantwoord. Er worden 3 ordes besproken: 3<sup>de</sup> orde filter, 4<sup>de</sup> orde filter en 5<sup>de</sup> orde filter. Onderstaande coëfficiënten werden bekomen, door gebruik te maken van de methode vermeld in de vorige paragraaf.

3 <sup>de</sup> orde coëfficiënten	4 <sup>de</sup> orde coëfficiënten	5 <sup>de</sup> orde coëfficiënten
1	1	1
-0,17606	-0,17606	-0,17606
-0,03591	-0,03591	-0,03591
-0,01711	-0,01711	-0,01711
	-0,01656	-0,01656
		-0,00028

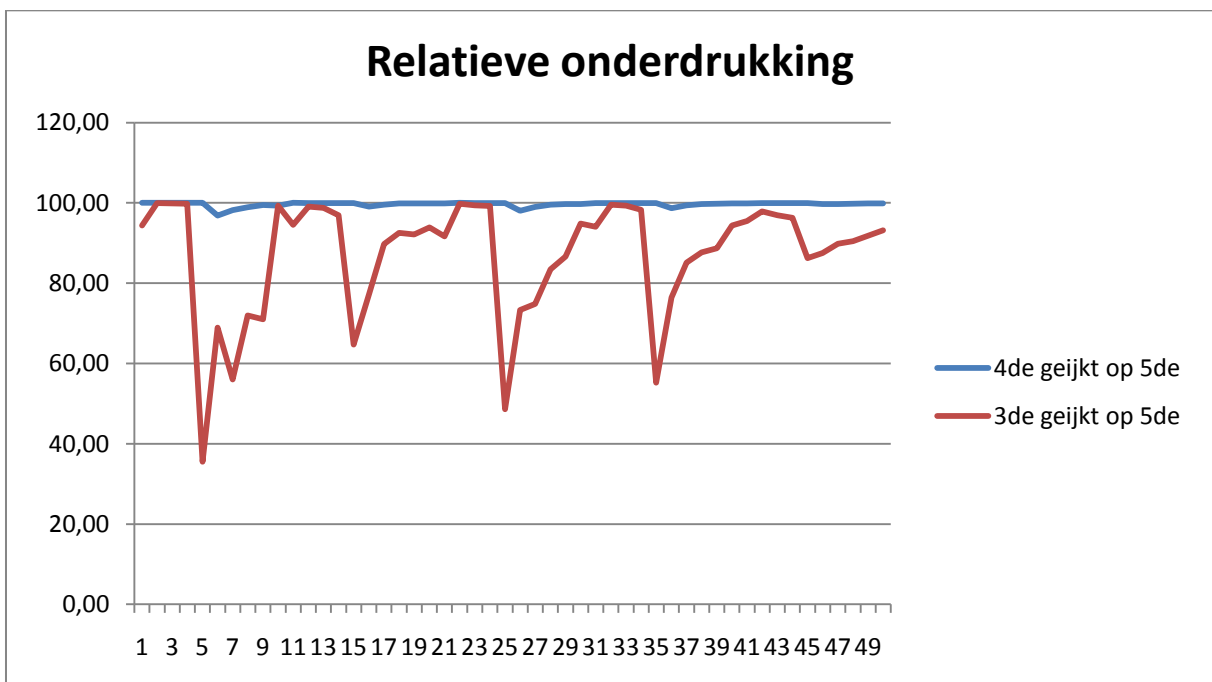
*Tabel 8: Filtercoëfficiënten*

Om de resultaten van de verschillende orde filters met elkaar te kunnen vergelijken, wordt gelineariseerde QIE-data aan de verschillende systemen aangelegd. Zo wordt onderstaande grafiek verkregen.



Bovenstaande grafiek zegt eigenlijk weinig. Er kan enkel gezegd worden dat de 3<sup>de</sup> orde filter bij signalen met hoge amplitudes na de puls opnieuw een piek vertoont. Daarnaast moet er ook opgemerkt worden dat wanneer de pulsform groter is dan de gemiddelde impulsresponsie, er overcompensatie van het signaal ontstaat.

Om een beter beeld te vormen van de verschillende onderdrukkingen wordt als volgt te werk gegaan: er wordt aangenomen dat de 5<sup>de</sup> orde filter de maximale onderdrukking levert. Per filter wordt de onderdrukking berekend door de oorspronkelijke data te verminderen met de gecorrigeerde waarde. Vervolgens wordt de relatieve onderdrukking berekend ten opzichte van het resultaat van de 5<sup>de</sup> orde filter. Zo wordt volgende grafiek verkregen.

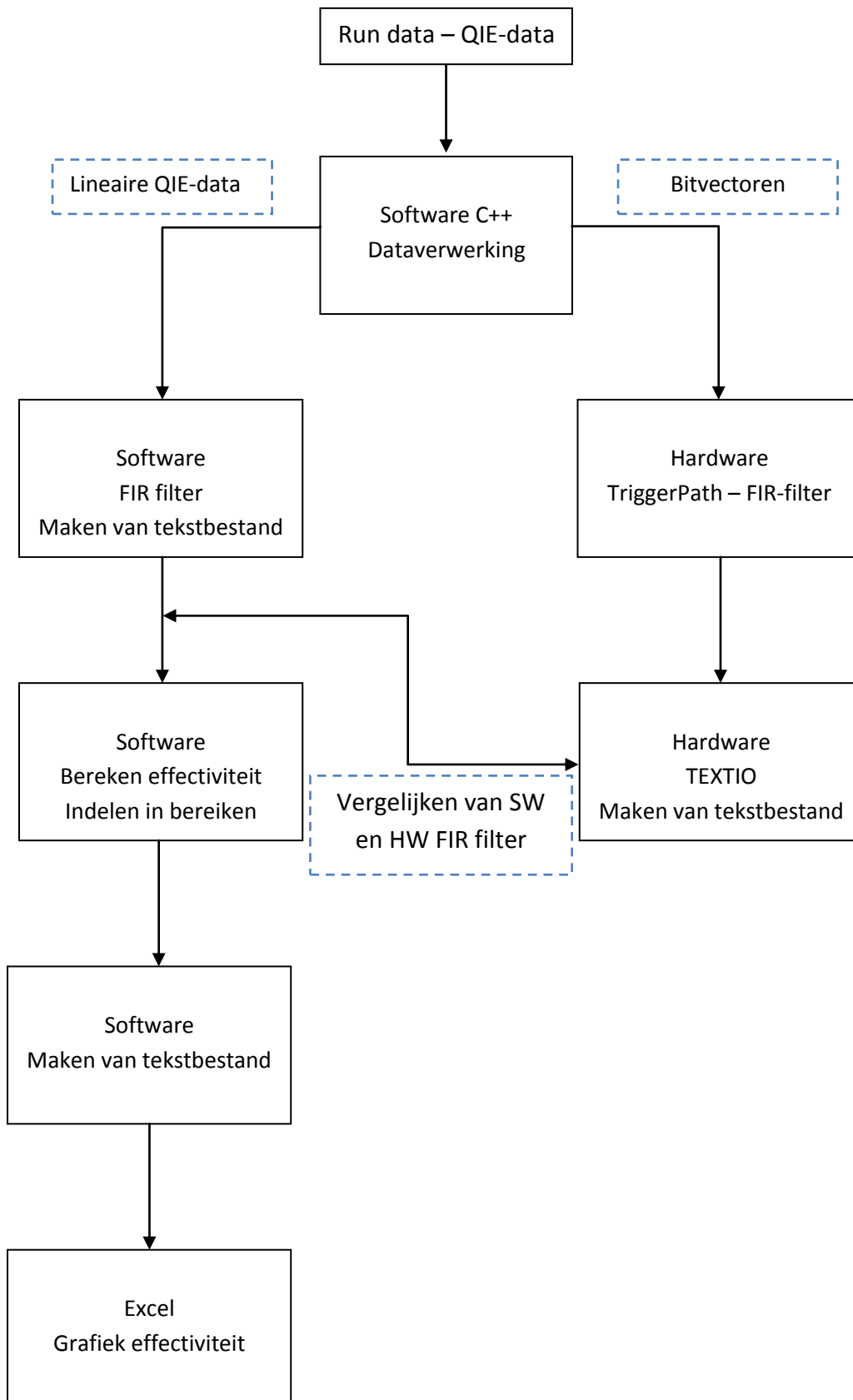


Aan de hand van vorige grafiek wordt het verschil tussen de verschillende orde filters duidelijk. Er kan duidelijk gezien worden dat de onderdrukking van de 3<sup>de</sup> orde filter 40% tot 60% lager ligt dan die van de 4<sup>de</sup> en 5<sup>de</sup> orde filter. Daarnaast kan er ook een klein verschil opgemerkt worden tussen de 4<sup>de</sup> en 5<sup>de</sup> orde filter van ongeveer 5 %.

#### **6.4.1 Besluit**

De 3<sup>de</sup> orde FIR filter heeft significant minder onderdrukking dan de 4<sup>de</sup> en 5<sup>de</sup> orde filter. Aan de hand van vorige grafieken is de 5<sup>de</sup> orde FIR filter de beste keuze. Er zal getracht worden de 5<sup>de</sup> orde filter te implementeren in hardware (zie hoofdstuk 8: Hardware). Indien de 5<sup>de</sup> orde filter te veel resources gebruikt, kan er teruggevallen worden op de 4<sup>de</sup> orde filter.

## 6.5 Methodiek voor het ontwerp van de filters



*Figuur 36: Methodiek*

Figuur 36: Methodiek beschrijft de manier waarop te werk is gegaan voor het simuleren van de filters. Omwille van het feit dat simulaties van hardware redelijk wat tijd innemen zeker wanneer het hardware project in grootte toeneemt, werd er besloten om ook een FIR filter te implementeren in software.

Onbewerkte CASTOR data van een run in 2010 wordt verwerkt door het programma. De tijd tussen twee botsingen in deze run is meerdere microseconden. Dit houdt in dat er geen overlapping is van twee opeenvolgende signalen wat deze data zeer bruikbaar maakt als uitgangspunt om het filter te ontwerpen en te beoordelen.

Om de hardware te simuleren, moet de niet-lineaire QIE-data geconverteerd worden naar een bitvector en dit per module van CASTOR. Per module wordt een tekstbestand aangemaakt waarin alle bitvectoren van de betreffende module worden in opgeslagen. Deze tekstbestanden worden ter beschikking gesteld aan het programma dat zorgt voor de hardwaresimulatie, namelijk Modelsim. De output van de filters wordt weggeschreven in een tekstbestand.

Daarnaast is het ook mogelijk om de software de rundata te laten lineariseren. Deze operatie is equivalent met de LUT's in de hardware. Na de linearisatie kan de data aangelegd worden aan een FIR filter in de SW. Net zoals bij de hardware moet hier ook een reeks coëfficiënten meegegeven worden. De output van de SW FIR filter wordt ook hier weggeschreven naar een tekstbestand. Nu kunnen de resultaten van de HW filter met de resultaten van de SW filter vergeleken worden. De resultaten komen goed overeen. De fout bedraagt gemiddeld 2 LSB's (=4) per filter. Een mogelijke oorzaak van deze afwijking kan zijn dat de deling nauwkeuriger gebeurt op een computer. Er wordt besloten dat de resultaten voldoende overeenkomen om de optimalisatie van het FIR filter met behulp van de software uit te voeren.

Ten slotte bevat de software ook de mogelijkheid om de effectiviteit van de filter te bepalen voor bepaalde amplitudebereiken. De output van deze functionaliteit zijn tekstbestanden waarin het gemiddelde residu van het signaal voor en na de correctie in wordt opgeslagen. Meer informatie hiervoor in hoofdstuk 7: Software.

## 7 Software

Het doel van de software tijdens dit project is voornamelijk dataverwerking. Om VHDL-simulaties te kunnen uitvoeren moeten decimale getallen omgezet worden in binaire vectoren met een bepaalde lengte. Daarnaast is het ook mogelijk om een FIR filter in software te implementeren. De software implementatie van de digitale filter zorgt ervoor dat er tijd gewonnen wordt tijdens het ontwerpproces. De gebruikte software IDE is Code::Blocks, hierbij wordt de GNU GCC compiler gebruikt.

### 7.1 Dataverwerking met C++

Om bewerkingen op de QIE-data te kunnen uitvoeren moet deze data eerst omgezet worden in lineaire data. De QIE-data wordt uitgelezen uit tekstbestanden. Elke lijn begint met het eventnummer. Daarna staat er de kanaalnummer met vervolgens de eventdata. Eventdata zijn telkens 10 samples die genomen zijn met een interval van 25ns.

```
evt:3986577 ch1: 6;6;7;6;87;50;33;22;18;4;
```

Per kanaal bestaat er zo'n tekstbestand. Voor het inlezen van de tekstbestanden wordt er gebruik gemaakt van ifstream en van de klasse StringTokenizer[15]. Een StringTokenizer is een klasse waaraan er een string moet meegegeven worden en een delimiter. Wanneer bovenstaande lijn wordt meegegeven en de delimiter wordt ingesteld op ";", dan zal de regel worden ingedeeld in verschillende substring: "evt:3986577 ch1: 6", "6", "7", "87", "50", "33", "22", "18", "4". De klasse is zeer handig bij het uitvoeren van stringmanipulaties.[16]

Voor elke module van CASTOR wordt er een object van de klasse Channel aangemaakt. Deze klasse houdt een vector van Qevents bij. De klasse Qevents bevat de eigenlijke data. De klasse bevat daarnaast ook de nodige functies voor het berekenen van een gemiddelde impulsresponsie en het uitvoeren van omzettingen: decimaal naar binair en niet-lineair naar lineair.

Voor de linearisatie van de data wordt er gebruik van de klasse CastorQIEShape. Deze klasse bevat een tabel die waarden bevat voor het inter- en extrapoleren van niet-lineaire data.

De klasse Out is verantwoordelijk voor het wegschrijven van data naar tekstbestanden. Hier wordt er gebruik gemaakt van constructor overloading. Dit maakt het mogelijk om de klasse te gebruiken in meerdere contexten. Zo kan de klasse een volledig kanaal wegschrijven naar een tekstbestand, een array wegschrijven (wordt gebruikt voor gemiddelde impulsresponsie) en een vector van Qevents wegschrijven.

De klasse ConvertToDec zorgt voor de omzetting van binaire vectoren naar decimale getallen. De reden voor het ontstaan van deze klasse is het feit dat de TEXTIO-library in VHDL geen integer kan wegschrijven naar een tekstbestand, enkel bitvectoren. De omzetting van

binair naar decimaal zorgt ervoor dat output van de hardware filter vergeleken kan worden met de filter in software.

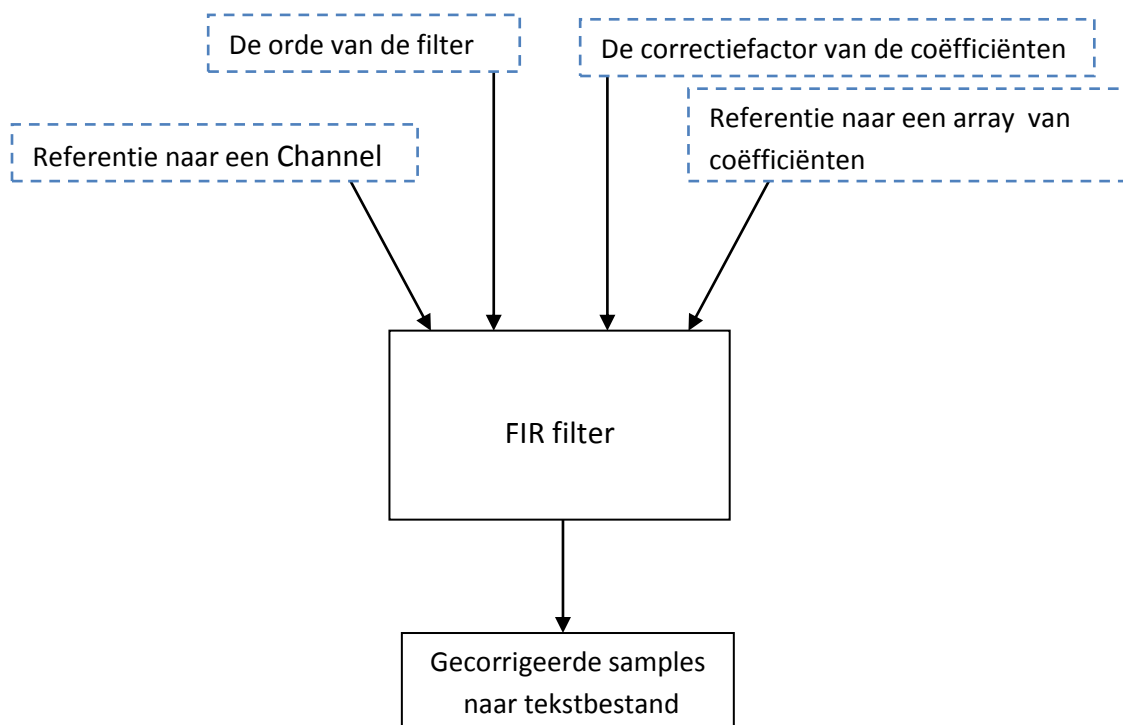
Ten slotte moet bij het ontwerpen van de klasse ConvertToBin rekening gehouden worden dat de binaire vectoren als signed vectoren geïnterpreteerd moeten worden. Zo kan deze klasse gebruikt worden om negatieve filtercoëfficiënten te converteren.

## 7.2 Software implementatie van een FIR filter

Voor de implementatie van een FIR filter in software, is er een klasse FIR geschreven.

De input dat deze klasse nodig heeft is:

- Referentie naar een reeks coëfficiënten
- Referentie naar een Channel d.w.z. een lijst met lineaire inputwaarden
- De orde van het filter
- De correctiefactor die doorgevoerd wordt in de hardware om een nauwkeurigere deling te verkrijgen.



*Figuur 37: Klasse FIR*

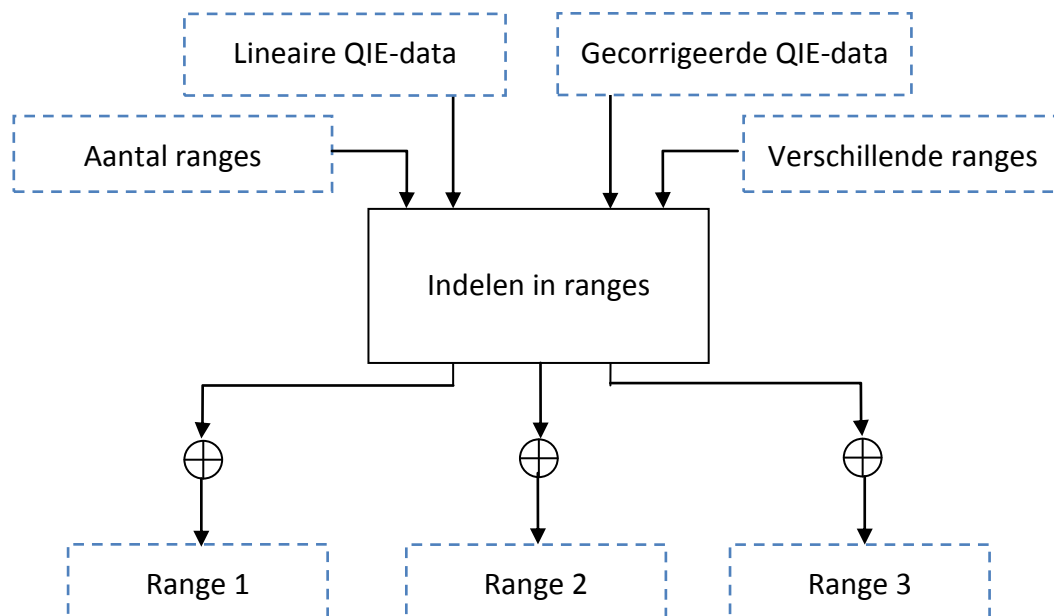
De klasse FIR bevat een algoritme dat de logische structuur van een FIR filter evenaart. Het algoritme onthoudt en verschuift de ingangswaarde. Hoeveel ingangswaarden er bijgehouden worden, is afhankelijk van de meegegeven orde. Elke tap wordt vermenigvuldigd met de overeenkomstige coëfficiënt. Telkens wanneer de filter een output genereert wordt deze waarde opgeslagen in een interne vector en wordt deze waarde

weggeschreven in een tekstbestand. De interne vector kan later gebruikt worden om de effectiviteit van de filter te bepalen. Het gegenereerde tekstbestand wordt gebruikt om grafieken te maken in Excel van de filterresultaten.

### 7.3 Bepalen van de effectiviteit van een filter

Op onderstaande figuur wordt de werking van de functie die de effectiviteit van een filter berekent schematisch voorgesteld. De functie heeft volgende inputwaarde nodig:

- Referentie naar Channel die de lineaire QIE-data bevat
- De interne gecorrigeerde lineaire QIE-data
- Het aantal bereiken
- Referentie naar een array van ranges.



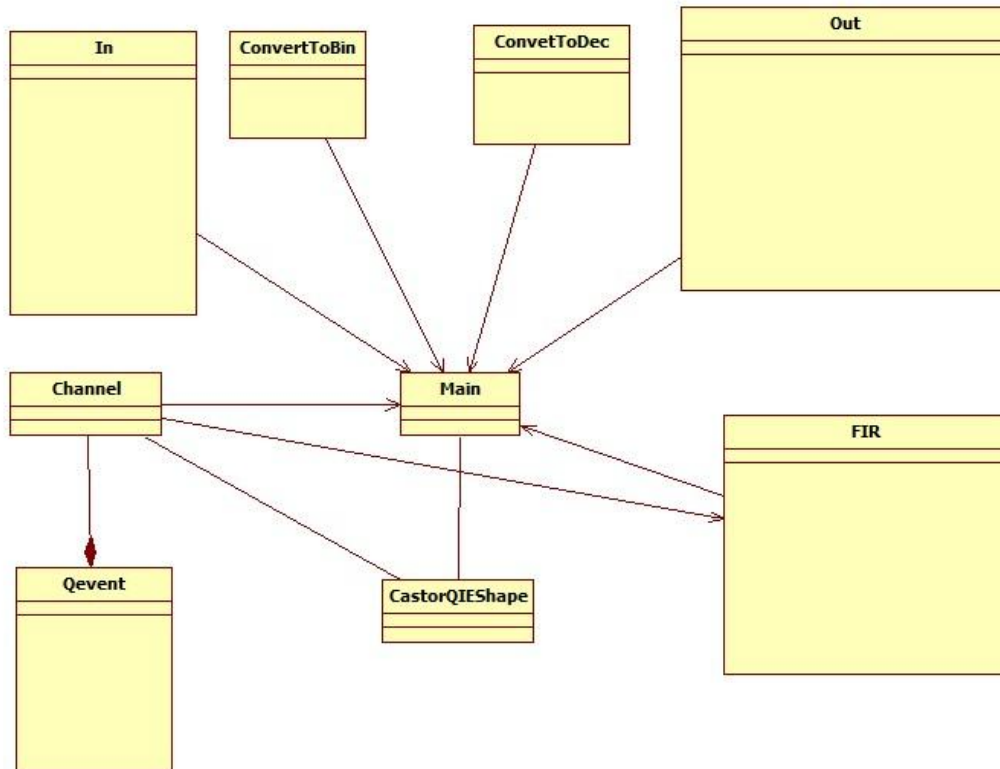
Figuur 38: Effectiviteit van een filter

Tijdens de initialisatie zullen array's gereserveerd worden en zullen alle waarden ervan op nul gezet worden. Vervolgens zal het algoritme elk event indelen in een bepaalde bereik. Na de indeling wordt het event gesommeerd met de array van zijn bereik. Per bereik wordt er een teller bijgehouden dat weergeeft hoeveel events er aan een bepaalde bereik zijn toegevoegd.

Wanneer alle events verwerkt zijn, worden elke waarde van de array van een bereik gedeeld door zijn overeenkomstige teller. Op deze manier wordt het gemiddeld residu bepaald voor en na de correctie.

## 7.4 UML-diagram

Onderstaand UML-diagram toont de verschillende relaties tussen de verschillende klassen. Zo bestaat de klasse Channel uit een aantal object van de klasse Qevent. Er bestaat een relatie tussen de klasse Channel en de klasse CastorQIEShape. Deze klasse wordt gebruikt voor het lineariseren van de QIE-data. De klasse Qevent bevat zowel de lineaire als de niet-lineaire data. De klasse Channel bevat alle Qevents en kan alle mogelijke functionaliteiten delegeren.



Figuur 39: UML-diagram

## 8 Hardware

De FPGA<sup>19</sup> die op de HTR-kaart is geplaatst, is de Virtex2 XC2V3000 BF957 van Xilinx. Voor meer informatie over de gebruikte FPGA wordt er verwezen naar de datasheet[20] en de userguide[21].

### 8.1 Gebruikte software



#### 8.1.1 Modelsim SE 6.4a

Modelsim is een softwarepakket waarmee hardware kan gesimuleerd worden voordat de eigenlijke implementatie plaatsvindt. Door het simuleren van de hardwarearchitectuur kunnen fouten vroegtijdig opgemerkt worden en verbeterd worden. Het software platform verleent hierbij een zeer belangrijke rol.

Het triggerpath is reeds geschreven in Verilog. Verilog is net zoals VHDL een taal waarmee de gedragingen van hardware kan worden beschreven. Voor meer informatie wordt er verwezen naar hoofdstuk 8.2.

Voor het uitvoeren van een gemengde simulatie d.w.z. dat er zowel verilog-modules als VHDL-componenten aanwezig zijn in de architectuur van het project, moet er gewerkt worden met een professionele versie van Modelsim. Andere versie zoals de student-versie, hebben die functionaliteit niet.



#### 8.1.2 ISE 9.1

ISE is een softwarepakket dat de mogelijkheid biedt om hardware te synthetiseren en te implementeren op een FPGA. Het synthetisatie-proces gaat kijken of het mogelijk is om bepaalde code-structuren uit te voeren.

Tijdens de implementatiefase worden de verschillende logic-cells zodanig geconfigureerd dat deze de FPGA de opgelegde functionaliteit krijgt.

## 8.2 VERILOG & VHDL

### 8.2.1 Verilog

Verilog is een hardware description language. Het wordt gebruikt voor het modelleren van elektronische systemen. De syntax van verilog is te vergelijken met die van C/C++.

### 8.2.2 VHDL

VHDL staat voor VHSIC (Very High Speed Integrated Circuit) Hardware Description Language. Het is een taal waarmee hardware beschreven kan worden. Het schrijven van hardware laat toe om gemakkelijk wijzigingen aan te brengen. Daarnaast kan de code ook uitvoerig getest

---

<sup>19</sup> FPGA: Field-Programmable Gate Array

en gesimuleerd worden. In tegenstelling tot Verilog maakt VHDL gebruik van een vast codestructuur. IEEE<sup>20</sup> heeft de standaardisatie van VHDL tot stand gebracht.

### 8.2.3 Samenwerking VHDL & Verilog

VHDL en Verilog hebben beide hun eigen manier voor het instantiëren van hardware componenten. Het verschil tussen de manier waarop verschillende hardwarecomponenten geïntanceerd moeten worden, is weergegeven in onderstaande tabel. In de linkerkolom wordt er een voorbeeld gegeven van hoe een component in VHDL geïmplementeerd moet worden. In de architectuur van de hogere component moet de componentdeclaratie aanwezig zijn. Vervolgens wordt in de body van de architectuur een port map gemaakt die de component aanstuurt.

In Verilog is de instantiatie eenvoudiger. Zowel VHDL-componenten als Verilog-modules moeten op dezelfde wijze geïmplementeerd worden. Een voorbeeld hiervan is terug te vinden in de rechterkolom van onderstaande tabel.[2]

VHDL	Verilog
<pre>//Component declaratie component MULT18X18 port( A : in std_logic_vector (17 downto 0); B : in std_logic_vector (17 downto 0); P : out std_logic_vector (35 downto 0) ); end component;  //Architectuur U_MULT18X18 : MULT18X18 port map ( A =&gt; , -- insert input signal #1 B =&gt; , -- insert input signal #2 P =&gt; -- insert output signal );</pre>	<pre>//Instantatie MULT18X18 U_MULT18X18 ( .A () , // insert input signal #1 .B () , // insert input signal #2 .P () // insert output signal );</pre>

Tabel 9: Vergelijking tussen VHDL en Verilog

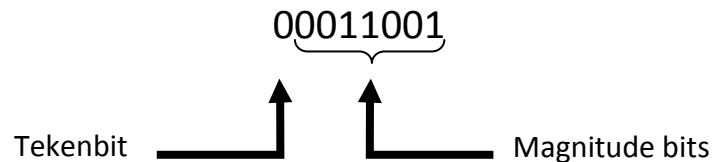
## 8.3 Signed getallen

Signed getallen[10] worden in de digitale wereld gebruikt om negatieve getallen te representeren. De meest gebruikte vorm is 2's complement. Bij het gebruik van signed getallen kan in het algemeen aangenomen worden dat de MSB de tekenbit voorstelt. Indien de tekenbit gelijk is aan '1', dan representeert het binair getal een negatief decimaal getal. Is

<sup>20</sup> IEEE: is een internationale vereniging van ingenieurs van verschillende vakgebieden.

de tekenbit gelijk aan '0', dan stelt dit binair getal een positief decimaal getal voor. Voorbeeld: een 3-bit getal heeft een range van 0 ("000") tot 7 ("111"). Een signed 3-bit getal heeft volgend bereik vanwege de tekenbit. Het bereik van dit getal zal zijn: -4 ("100") tot 3 ("011").

De andere bits worden beschouwd als magnitude bits. Deze bepalen de grote van het getal ongeacht het teken.



*Figuur 40: Signed getal*

### 8.3.1 Conversie van binair naar 2's complement

Om een binair getal te converteren naar de 2's complement vorm moet het getal eerst worden omgezet in de 1's complement vorm. Dit gebeurt door alle bits te inverteren. Vervolgens kan de 2's complement bekomen worden door bij de 1's complement één bij op te tellen.

$$N = 6 = 0110 \Rightarrow \bar{N} = -6 = 1001(1's\ complement)$$

$$\bar{N} = -6 = 1001 + 1 = 1010(2's\ complement)$$

## 8.4 FIR

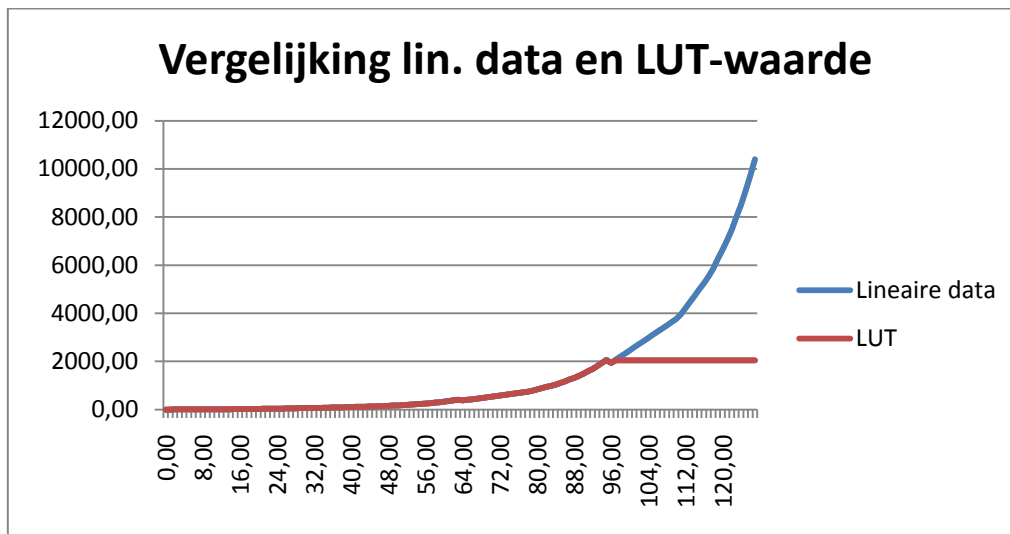
De logica op de FPGA's van de HTR-kaart werd reeds geïmplementeerd in Verilog. In samenspraak met mijn promotor is er beslist om de filter te implementeren in VHDL daar men vertrouwd is met deze taal.

In hardware kan een FIR filter gerealiseerd worden door gebruik te maken van shiftregisters, vermenigvuldigers en optellers. De nodige aantallen van deze componenten wordt bepaald door de orde van de filter. Hoe hoger de orde van de filter, hoe groter het aantal componenten nodig voor de realisatie.

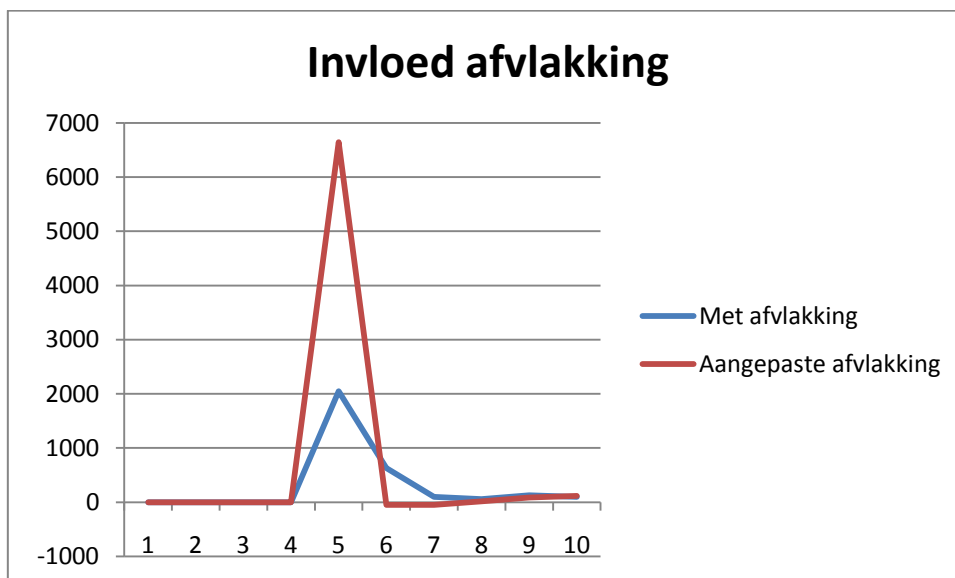
## 8.5 LUT

De LUT is verantwoordelijk voor de conversie van niet-lineaire getallen naar lineaire getallen. De LUT's zijn 11 bit breed. De 7 bit QIE-data (2 bit exponent + 5 bit mantissa) wordt gebruikt om het adres van de LUT te bepalen. De niet-lineaire QIE-data heeft een range van 0 tot 127. Indien deze data wordt omgezet naar de lineaire vorm, heeft de data een bereik van 0 tot 10396. Wat direct opvalt is dat 10396 niet voorgesteld kan worden door 11 bit ( $2^{11}=2047$ ).

Oorspronkelijk werd deze data afgevlakt. De fout die dan gemaakt wordt, wordt weergegeven in onderstaande grafiek.



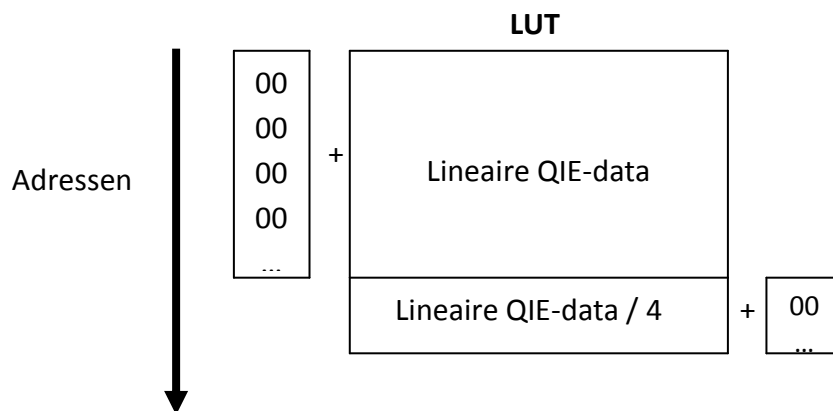
Wanneer de QIE-data gelijk is aan 96, wordt de data sterk afgevlakt. Voor de filters vormt dit een probleem. Aangezien de coëfficiënten van de filters berekend worden aan de hand van de impulsresponsie, is het van groot belang dat deze zo goed mogelijk wordt benaderd. Indien er nu een event met een hoge amplitude beschouwd wordt, dan wordt de piek afgevlakt. Terwijl de rest van de data hetzelfde blijft. Hierdoor verandert de responsie van de puls. De filter zal niet meer op dezelfde manier kunnen onderdrukken. De grafiek hieronder moet duidelijk maken wat de impact is van afvlakking.



Om de afvlakking tegen te gaan is volgende systeem bedacht. De lineaire data gaat van 0 tot 10396. Indien het bereik volledig voorgesteld wordt, is een getal nodig van 14 bit. Het maximale bereik van dit getal is gelijk aan 16384. Het bereik van een 14 bit getal is dus ongeveer 6000 eenheden te groot. Om een optimaler resourcegebruik te bekomen, wordt er

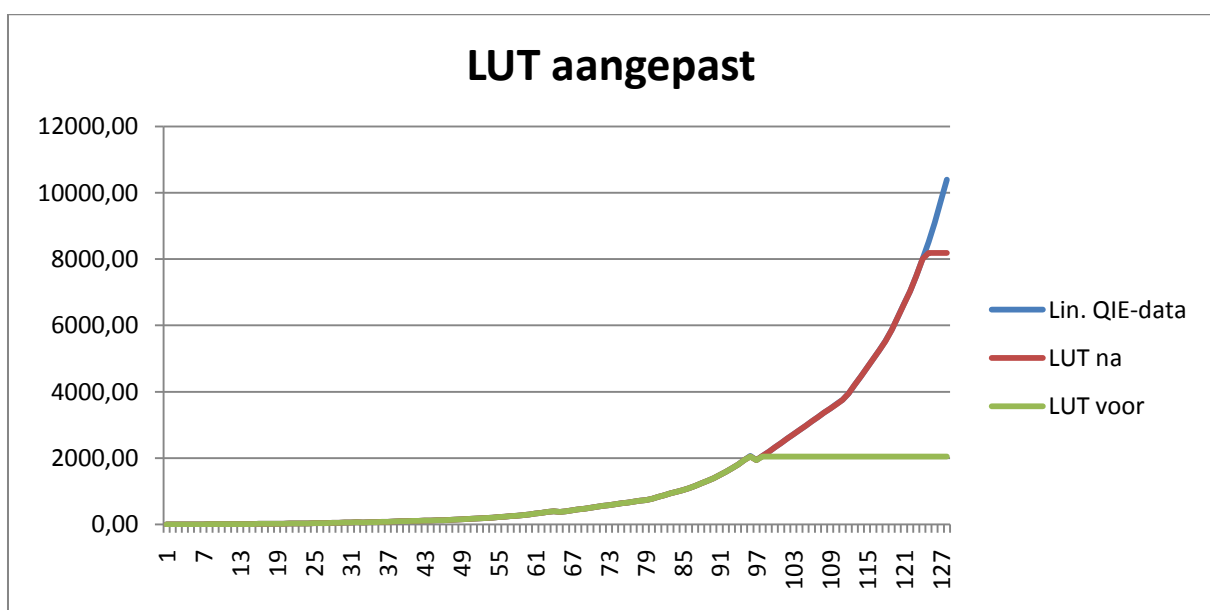
gekozen voor een 13 bit getal met een bereik van 8192. Dit getal zorgt voor een kleine afvlakking.

Om vervolgens de lineaire QIE-data toch te kunnen voorstellen met 11 bit, wordt onderzocht bij welke QIE-data de lineaire waarde groter wordt dan 2047. De waarde van de QIE-data waarbij dit optreedt is gelijk aan 96. Met dit in het achterhoofd, wordt het volgende gedaan. De signalen van de QIE dienen voor de adresbepaling van de LUT. Wanneer de QIE-data kleiner is dan 96 wordt er niets ondernomen. Echter wanneer deze groter wordt de lineaire QIE-data gedeeld door 4.



Figuur 41: Aanpassing LUT

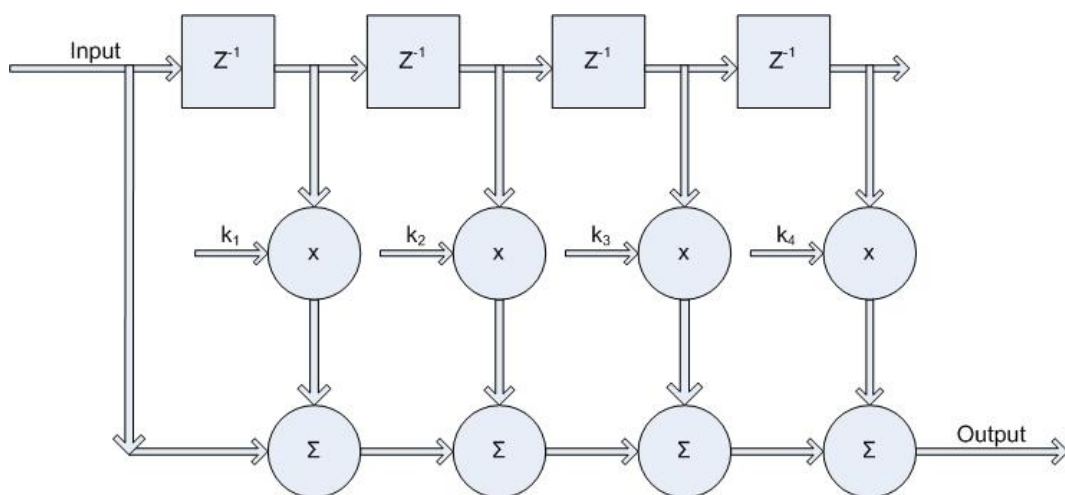
Wanneer de lineaire waarde vanuit de LUT komt, weet de component die de LUT aanstuurt, welk adres er is aangelegd. Deze component gaat dit adres gebruiken om te bepalen welke actie er ondernomen moet worden. Indien de aangelegde QIE-data kleiner is dan 96, wordt er vooraan de bitvector twee nullen toegevoegd. Is de QIE-data groter dan of gelijk aan 96, dan worden er achteraan de bitvector twee nullen toegevoegd.



## 8.6 Optimalisatie van hardwarecomponenten

In hardware kan een vermenigvuldiging op twee manier gebeuren. De eerste mogelijkheid waarmee een vermenigvuldiging kan worden uitgevoerd is een multiplier. Dit is de meest efficiënte manier om deze bewerking uit te voeren. Daarnaast is het ook mogelijk door middel van een LUT een vermenigvuldiging uit te voeren op voorwaarde dat een van beide factoren constant is. De FPGA bevat 96 embedded multipliers. Om tijdens de synthetisatiefase de hardware te verplichten deze componenten te gebruiken wordt deze component geïnstantieerd en geïmplementeerd in de component FIR.

Een tweede optimalisatie kan doorgevoerd worden, nadat de filtercoëfficiënten beter bestudeerd werden. Indien er gekeken wordt naar de filtercoëfficiënten, valt het op dat de eerste coëfficiënt telkens gelijk is aan 1. In plaats van de huidige input waarde aan te leggen aan een multiplier, kan deze dus direct doorgesluisd worden naar de sommatie. Deze optimalisatie bespaart één multiplier per filter. De structuur van de geoptimaliseerde FIR filter wordt voorgesteld in onderstaande figuur.

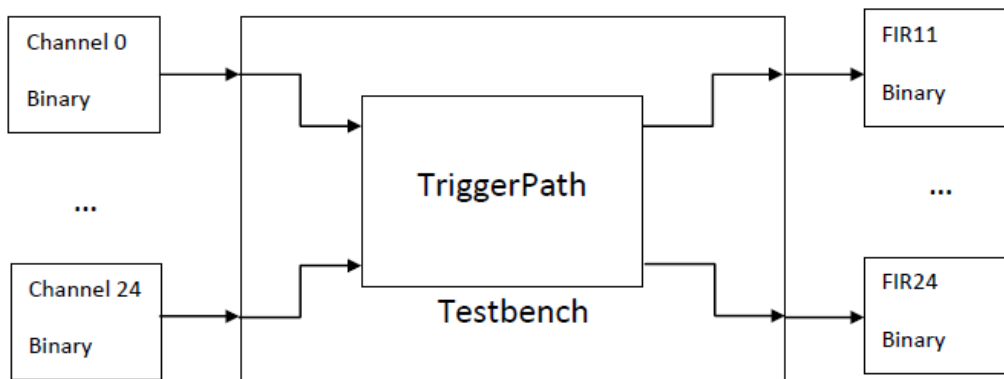


*Figuur 42: Geoptimaliseerde FIR filter*

Het is belangrijk om de FIR filter te optimaliseren op het gebruik van embedded multipliers omdat elke multiplier geassocieerd wordt met een 18kBit SelectRAM. Deze RAM blokken worden ook gebruikt als LUT. Naar de toekomst toe, wordt op deze manier de mogelijkheid gecreëerd om nog extra LUT toe te voegen. Daarnaast verwacht de multiplier 18 bit getallen aan de input. De LUT-data en de coëfficiënten zijn echter kleiner. Er zal dus extra logica voorzien moeten worden om de nodige bits toe te voegen. De output van de multipliers zal een 36 bit getal zijn. Om de nauwkeurigheid van de filter te verhogen, wordt elke coëfficiënt met een bepaalde factor vermenigvuldigd. Deze factor kan gecorrigeerd worden door een aantal LSB's te laten vallen. Op deze manier hoeft er geen extra deling te gebeuren in de hardware.

## 8.7 Simulatie met de TEXTIO-library

De TEXTIO-library maakt het mogelijk om tekstbestanden in te lezen en weg te schrijven. Tijdens de simulatie wordt een tekstbestand van een bepaald kanaal regel per regel ingelezen. In het tekstbestand staat op elke regel één binaire inputwaarde van dat kanaal. Vervolgens wordt de binaire data naar het formaat bitvector geconverteerd omdat de TEXTIO-library enkel kan werken met dit formaat. Omwille van het feit dat er in de VHDL-code gewerkt met het formaat `STD_LOGIC_VECTOR` is er een conversie nodig. Vervolgens wordt de inputwaarde aangelegd aan het TriggerPath.



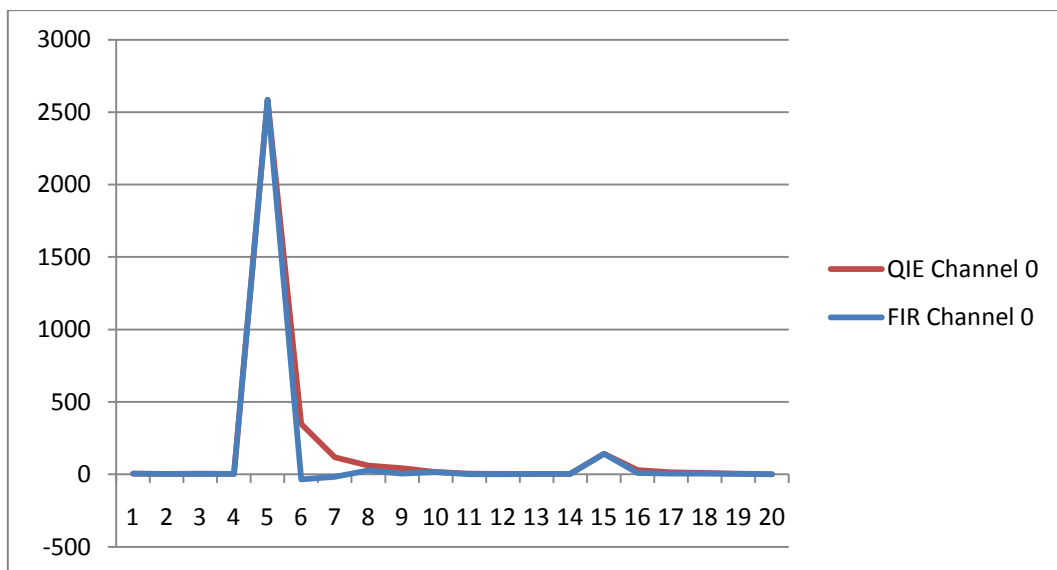
*Figuur 43: Schema simulatie*

Nadat het TriggerPath al de nodige bewerkingen heeft uitgevoerd, wordt de uitgang van de filter doorgestuurd naar de testbench. Van hieruit zal de output-data van elke filter binair worden weggeschreven naar de tekstbestanden gebruikmakend van de TEXTIO-library. De binaire inhoud van deze tekstbestanden kan vervolgens worden omgezet naar decimale getallen met behulp van de klasse `ConvertToDec`. Na de conversie is het mogelijk om de simulatie resultaten te vergelijken met de filterresultaten bekomen met behulp van software die in de volgende paragraaf wordt besproken.

## 8.8 Effectiviteit van het filter

Om de effectiviteit van het filter te bepalen, wordt de ingaande data vergeleken met de uitgaande data van het filter. Dit kan zowel toegepast worden op de SW filter als op de HW filter. Een voorbeeld hiervan wordt weergegeven in onderstaande grafiek. In deze grafiek stelt de rode curve de originele gelineariseerde QIE-data. De signaalstarten zijn hierbij duidelijk te zien. De uitgang van het filter wordt de blauwe curve weergegeven. In paragraaf 7.3: Bepalen van de effectiviteit van een filter staat meer informatie over hoe de effectiviteit van het filter wordt berekend.

Er is duidelijk te zien dat de signaalstart onderdrukt wordt. Wel dient er opgemerkt te worden dat wanneer de responsie afwijkt van de berekende gemiddelde impulsresponsie, de onderdrukking van de filter varieert. Bij grote signalen is de responsie sneller dan de gemiddelde responsie. In dat geval treedt er overcompensatie op en ontstaat een negatief signaal. Indien de responsie langzamer is dan de gemiddelde responsie, dan zal er te weinig onderdrukking zijn. Dit valt te verklaren aan de hand van de structuur van de FIR filter. Het is echter zo dat de onderdrukking op een bepaald moment afhangt van de x aantal vorige inputwaarde met x gelijk aan de orde van het filter. De huidige inputwaarde zal x aantal keer gebruikt worden om het signaal te corrigeren.



Bovenstaande grafiek is weinigzeggend. Voor grote signalen is de onderdrukking zichtbaar, voor kleine echter niet. Daarom zullen de residuen onderzocht worden voor verschillende amplitudebereiken. Om een beter beeld te krijgen op de onderdrukking van de FIR filter wordt in de volgende paragrafen het signaal ingedeeld in vier bereiken, namelijk: 0-300, 300-1000, 1000-3000, 300- 8188<sup>21</sup>.

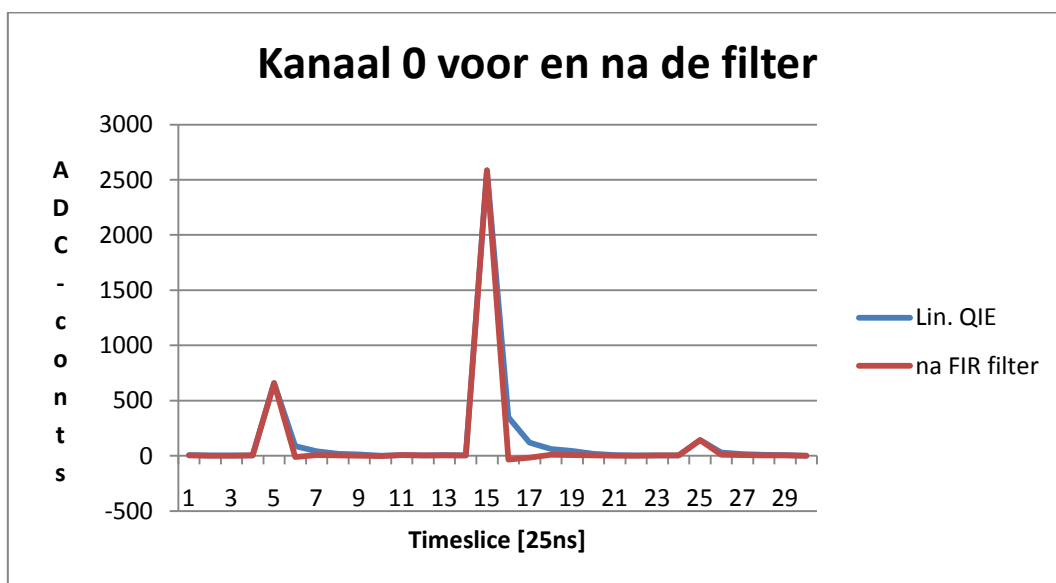
<sup>21</sup> Maximale output van aangepast LUT =2047\*4.

## 8.9 Implementatie van de FIR-filter

### 8.9.1 Implementatie per QIE-kanaal

Deze implementatie voorziet voor elk QIE-kanaal een FIR filter. Na synthetisatie van het project blijkt dat deze implementatie 119 % van de slices op de FPGA gebruikte. Dit percentage is te groot. Hierdoor zal deze implementatie niet gerealiseerd kunnen worden.

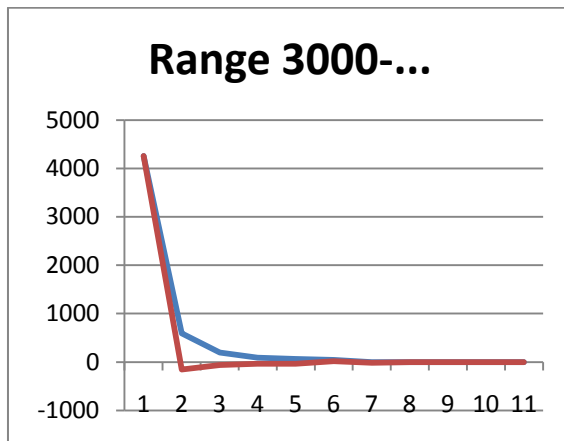
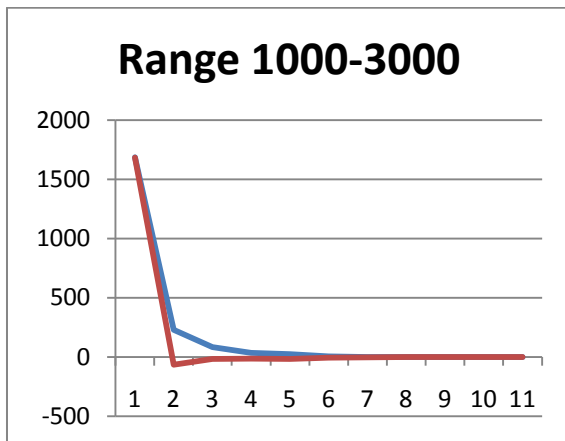
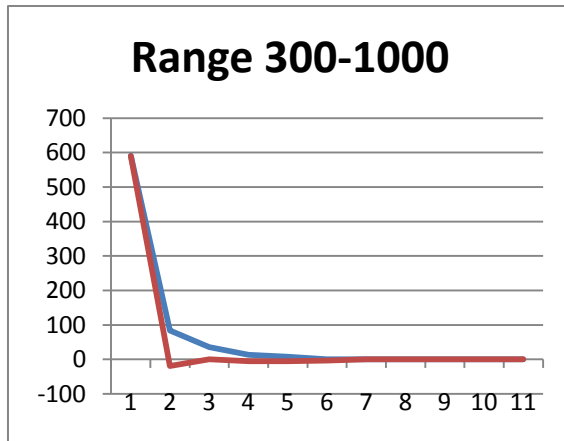
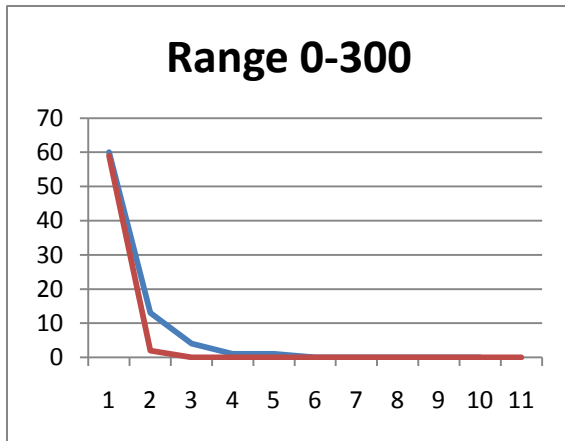
Ter volledigheid wordt hier toch nog een QIE-kanaal besproken. Voor het bepalen van de filtercoëfficiënten wordt gebruik gemaakt van de gemiddelde impulsresponsie van één kanaal. De methode voor het bepalen van de coëfficiënten werd reeds besproken in paragraaf 6.3. In onderstaande grafiek is het resultaat van dergelijke implementatie te zien.



Figuur 44: Resultaat implementatie per QIE

Om een beter inzicht te krijgen op de effectiviteit van de filter, kan er gekeken worden naar het resterende residu voor elke amplitudebereik. Er kan worden besloten dat de filter een goede onderdrukking heeft indien het signaal gelijk is aan de gemiddelde impulsresponsie. Kleine signalen hebben nauwelijks onderdrukking. Dit is mede het gevolg van afrondingsfouten. Voor kleine signalen zijn afrondingen meer significant dan voor signalen met een hogere amplitude. Daarnaast hebben grote signalen een heel sterke onderdrukking maar een oscillatorisch verloop.

In volgende tabel worden de verschillende gemiddelde residuen per bereik geïllustreerd. De onderdrukking voor het eerste bereik is niet ideaal. Voor de andere bereiken moet worden vastgesteld dat er overcompensatie optreedt.

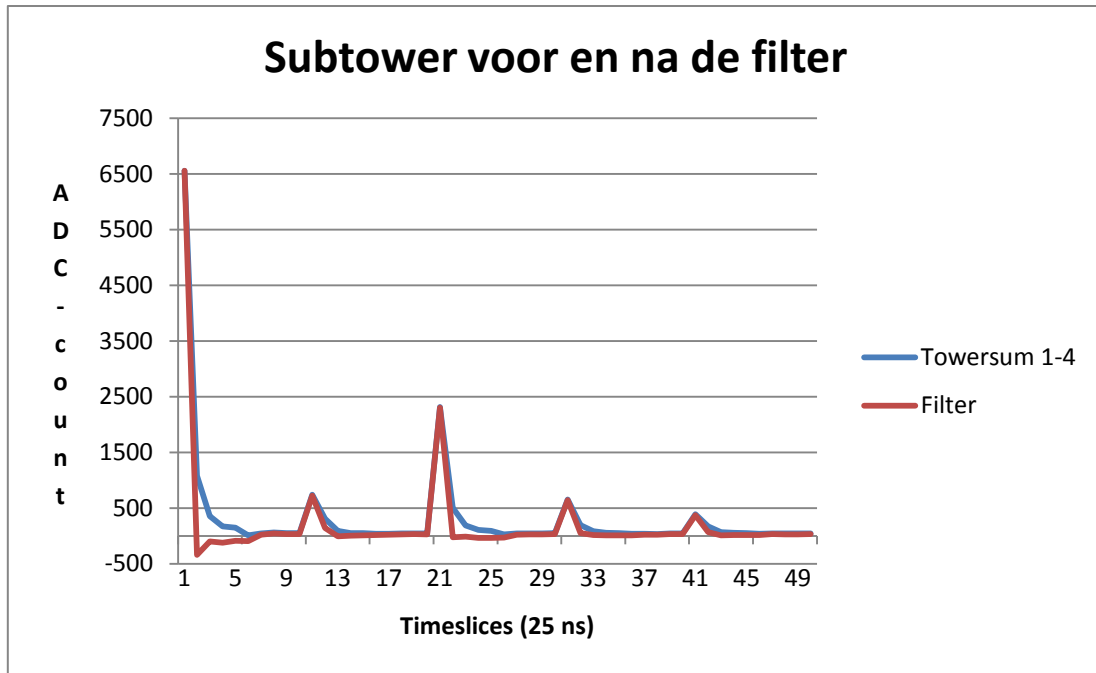


Tabel 10: Residubepaling per bereik – per QIE-kanaal

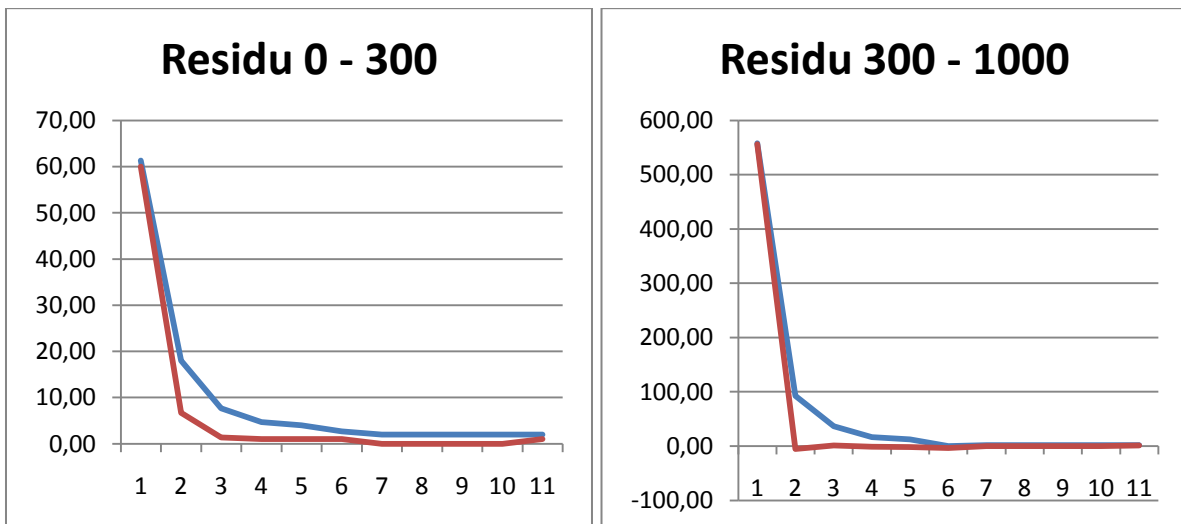
### 8.9.2 Implementatie per subtower

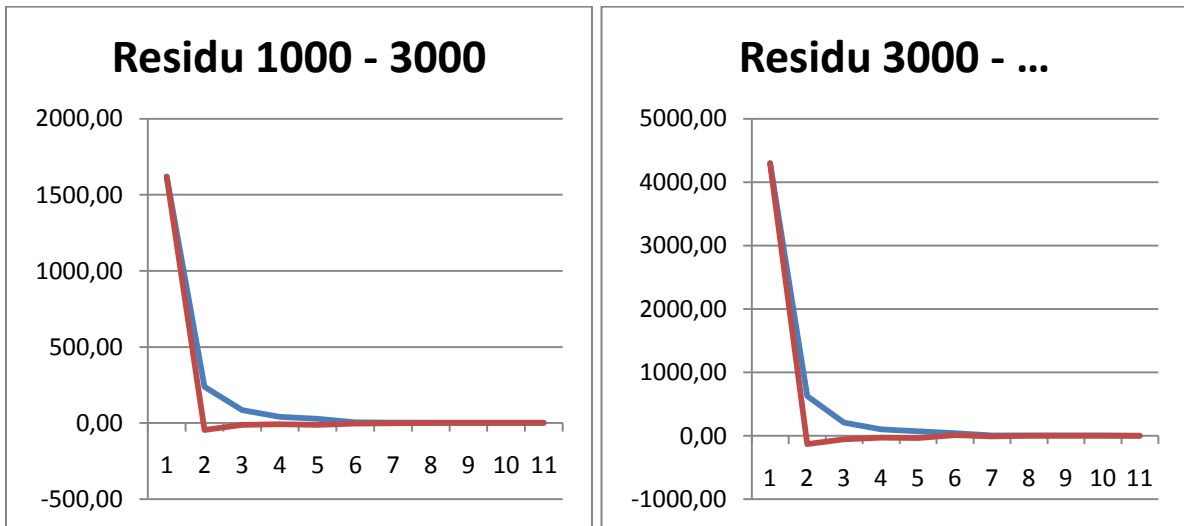
Bij deze implementatie wordt er per 3 QIE-kanalen één FIR filter geïmplementeerd d.w.z. dat de energie van drie modules wordt gecorrigeerd. Per triggertower worden er 4 FIR filters geïmplementeerd. Het moet duidelijk zijn dat dit gaat leiden tot een veel kleinere bezetting van de FPGA. Wanneer deze implementatie gesynthetiseerd wordt, wordt een bezetting van 85% verkregen. Dit is een aanvaardbaar percentage.

Op volgende figuur wordt de oorspronkelijk data en de gecorrigeerde data geplot.



Om een beter inzicht te krijgen op de effectiviteit van het filter, wordt ook hier het residu van het signaal bekeken voor vier schillende bereiken. De blauwe curve stelt het gemiddelde residu voor zonder de correctie. De rode curve representeert het gemiddelde residu na correctie door de FIR filter. Het gemiddelde residu voor de vier verschillende bereiken is weergegeven in Tabel 11: Residubepaling per bereik – per subtower.



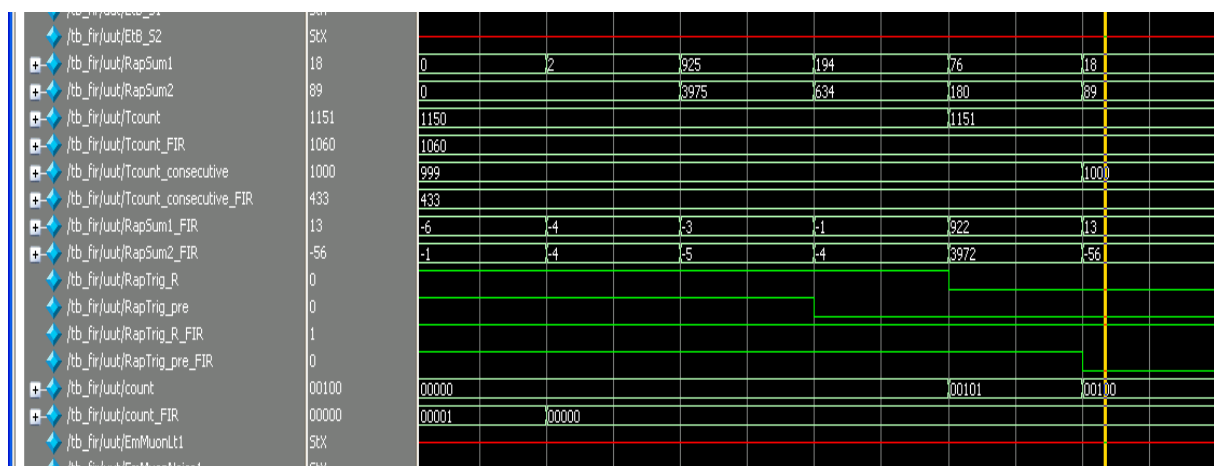


*Tabel 11: Residubepaling per bereik – per subtower*

## 9 Triggerresultaten

In de vorige paragraaf werd de onderdrukking van het filter en de resterende residuen uitvoerig besproken. Wat echter belangrijker is, is de invloed dat het filtersysteem heeft op de triggergeneratie. De QIE-staarten zorgen ervoor dat het signaal bijna nooit onder een bepaalde grenswaarde blijft. De trigger die geactiveerd wordt wanneer de energie-inhoud onder een bepaalde waarde blijft, is de NoLevelTrigger. Aan de hand van deze trigger zal onderzocht worden, in welke mate de filters de triggergeneratie verbeteren. In de simulatieomgeving Modelsim worden de nodige wijzigingen aan het TriggerPath aangebracht zodat de triggergeneratie voor de correctie vergeleken kan worden met de genereerde triggers na de correctie.

Om een idee te geven hoeveel triggers er minder worden gegenereerd, wordt de code aangepast. Aan de hardware worden tellers toegevoegd die samples en triggers tellen. De piek van het event zal geen Noleveltrigger geven. Dit zal resulteren in een nul op de triggeruitgang. Deze waarde wordt gebruikt om de tellers te synchroniseren. Wanneer er een nul op de triggeruitgang is geweest, wordt naar de volgende waarden gekeken. Wanneer er na de piek, opnieuw een nul op de triggeruitgang komt te staan, was de onderdrukking van de filter niet genoeg. In het andere geval wel.



Figuur 45: Triggerresultaat timeslice 6

Bovenstaande figuur bevat het triggerresultaat voor de 6<sup>de</sup> timeslice. In de simulatie werd er gewerkt met de waarde 19 als de grenswaarde voor de NolevelTrigger. De twee belangrijkste signalen op deze figuur zijn: Tcount\_consecutive en Tcount\_consecutive\_FIR. Deze twee tellers geven weer hoeveel maal de Noleveltrigger gelijk is aan nul op de sample na een piek. De Tcount\_consecutive bedraagt 1000, terwijl de Tcount\_consecutive\_FIR 433 bedraagt. Er worden dus 56,7%<sup>22</sup> minder triggers veroorzaakt dankzij het filtersysteem.

<sup>22</sup>  $\frac{1000-433}{1000} * 100 \% = 56,7\%$

Er werd net gekeken naar de triggergeneratie op de 6<sup>de</sup> timeslice van het event. Er kan ook onderzocht worden wat de invloed is op de 7<sup>de</sup> timeslice. Deze timeslice is op dit moment belangrijker dan de 6<sup>de</sup> timeslice omwille van het feit dat het tijdsinterval tussen de botsing van twee protonwolken 50ns bedraagt<sup>23</sup>. Op onderstaande figuur kan gezien worden dat de triggergeneratie van het verbeterde systeem gelijk is aan 117, terwijl bij het ongecorrigeerde systeem deze 1000 bedraagt. Dit is een verbetering van 88,3%<sup>24</sup>.

/tb_fir/uut/ETB_S2	50X						
/tb_fir/uut/RapSum1	44	107	37	19	44	6	
/tb_fir/uut/RapSum2	30	411	146	59	30	1	
/tb_fir/uut/Tcount	1269	1268	1269				
/tb_fir/uut/Tcount_FIR	1163	1162			1163		
/tb_fir/uut/Tcount_consecutive	1102	1101		1102			
/tb_fir/uut/Tcount_consecutive2	1000	999			1000		
/tb_fir/uut/Tcount_consecutive_FIR	620	620				621	
/tb_fir/uut/Tcount_consecutive_FIR2	117	117					
/tb_fir/uut/RapSum1_FIR	6	1	360	39	6	3	
/tb_fir/uut/RapSum2_FIR	3	-2	2595	5	3	-3	
/tb_fir/uut/RapTrig_R	0						
/tb_fir/uut/RapTrig_pre	0						
/tb_fir/uut/RapTrig_R_FIR	0						
/tb_fir/uut/RapTrig_pre_FIR	0						
/tb_fir/uut/count	00011	00000	00101	00100	00011	00010	
/tb_fir/uut/count_FIR	00101	00000			00101	00100	
/tb_fir/uut/EmuMupL1	50X						

Figuur 46: Triggerresultaat timeslice 7

De onderdrukking van dit systeem kan verbeterd worden door de eerste filtercoëfficiënt te verhogen. Zo wordt er eigenlijk voor een overcompensatie gezorgd. Bij deze manier van werken dient er opgemerkt te worden dat de energie-informatie van CASTOR wordt vervormd. In CASTOR zal data weggefilterd worden en er zal een NolevelTrigger gegenereerd worden. Wanneer deze trigger bijdraagt tot een L1Accept, dan wordt de originele data verstuurd naar de HLT. Deze bezit een eigen filteralgoritme en zal vaststellen dat er toch energie in CASTOR aanwezig is en zal het event droppen.

Ten slotte kan er ook gewerkt worden met niet-lineaire filters d.w.z. dat de coëfficiënt van elke tap van de filter afhankelijk is van de inputwaarde. Deze methode werd binnen het kader van thesis niet verder onderzocht.

<sup>23</sup> Gedurende de komende vier jaar zal de LHC met een wolkafstand van 50ns opereren in plaats van de ontwerpsspecificatie van 25ns, die als uitgangspunt voor het ontwerp van dit filter is gebruikt.

<sup>24</sup>  $\frac{1000-117}{1000} * 100 = 88,3\%$

## 10 Besluit

Het filtersysteem zorgt op de eerste timeslice na de piek (6<sup>de</sup> timeslice) voor een Noleveltriggervermindering van 57%. Dit betekent dat 43% van de gegenereerde triggers nog niet correct zijn. Dit is ver van optimaal. Om de resultaten op deze timeslice te verbeteren kan er in de toekomst onderzoek gedaan worden naar de implementatie van niet-lineaire filters waarbij de coëfficiënten afhankelijk zijn van de inputwaarde. Door beperkingen van de CASTOR detector wordt echter niet verwacht dat CASTOR operationeel zal zijn in de bundel met een 25ns botsingsinterval.

De tweede timeslice na de piek ondervindt een goede onderdrukking. De verbetering van de Noleveltrigger op deze slice bedraagt 88,3%. Aan de hand van dit resultaat is het zinvol om de filters te implementeren, dit op voorwaarde dat het interval tussen de botsing van twee protonwolken gelijk blijft aan 50ns.

Voor de Energietrigger kan aan de hand van de grafieken in hoofdstuk 8 besloten worden dat hier een significante verbetering is. Het filtersysteem zorgt ervoor dat de energie van de QIE-staart zodanig onderdrukt wordt dat er zo goed als geen incorrecte Energietriggers meer plaatsvinden. Het zijn vooral de QIE-starten van signalen met grote amplitudes die hiervoor onderdrukt moeten worden.

## Lijst van tabellen

Tabel 1: CERN leden .....	1
Tabel 2: CERN waarnemers .....	2
Tabel 3: Kabelspecificatie .....	25
Tabel 4: Amplitude i.f.v. frequentie .....	28
Tabel 5: Amplitude i.f.v. frequentie DRAKA .....	29
Tabel 6: Resultaten vierpuntsmeting .....	31
Tabel 7: Vergelijking meting - model .....	37
Tabel 8: Filtercoëfficiënten .....	43
Tabel 9: Vergelijking tussen VHDL en Verilog .....	53
Tabel 10: Residubepaling per bereik – per QIE-kanaal .....	61
Tabel 11: Residubepaling per bereik – per subtower .....	63

## Lijst van figuren

Figuur 1: LHC .....	2
Figuur 2: Big Bang.....	3
Figuur 3: CMS detector.....	3
Figuur 4: Doorsnede CMS-detector .....	4
Figuur 5: CASTOR.....	6
Figuur 6: CASTOR voor- en zijaanzicht .....	6
Figuur 7: De werking van een PMT .....	7
Figuur 8: QIE pulsforming voor klein en groot signaal .....	8
Figuur 9: Conversietabel niet-lineaire ADC [22].....	9
Figuur 10: QIE-kaart .....	10
Figuur 11: HTR-kaart.....	10
Figuur 12: Data path.....	11
Figuur 13: Triggerarchitectuur .....	12
Figuur 14: Structuur HTR-kaart .....	13
Figuur 15: Structuur TriggerPath.....	13
Figuur 16: Transformatieschema .....	16
Figuur 17: Model .....	17
Figuur 18: Transformatieschema .....	21
Figuur 19: Dispersie op kabel 9m (geel) – 18m (paars).....	23
Figuur 20: Reflectie.....	24
Figuur 21: Rise time bepaling kabel 64ns.....	26
Figuur 22: Rise time bepaling kabel 8ns.....	26
Figuur 23: Rise time vergelijking 8ns - 16ns - 64 ns .....	27
Figuur 24: Sinus 1Hz en Amplitude 100mV .....	27
Figuur 25: Amplitude sinus 25MHz .....	28
Figuur 26: Opstelling fasemeting .....	30
Figuur 27: Opstelling vierpuntsmeting.....	31
Figuur 28: Curve-fit van amplitudekarakteristiek .....	33
Figuur 29: Responsie via basic fitting .....	34
Figuur 30: Het frequentiedomein .....	35
Figuur 31: Het tijdsdomein.....	35
Figuur 33: Model via transferfunctie .....	37
Figuur 34: Structuur FIR filter .....	39
Figuur 35: Signaalstroomschema .....	39
Figuur 36: Feedback filter.....	41
Figuur 37: Methodiek.....	46
Figuur 38: Klasse FIR.....	49
Figuur 39: Effectiviteit van een filter.....	50
Figuur 40: UML-diagram.....	51
Figuur 41: Signed getal.....	54

Figuur 42: Aanpassing LUT .....	56
Figuur 43: Geoptimaliseerde FIR filter .....	57
Figuur 44: Schema simulatie .....	58
Figuur 45: Resultaat implementatie per QIE.....	60
Figuur 46: Triggerresultaat timeslice 6.....	64
Figuur 47: Triggerresultaat timeslice 7.....	65

## Bibliografie

1. Phase, Group, and Signal Velocity.  
<http://www.mathpages.com/home/kmath210/kmath210.htm>, 2010.
2. Beniast A. *Elektronica-VHDL simulatie*. Hoboken: KDG.IWT; 2009.
3. CERN. CERN member states, 2008  
[https://ert.cern.ch/browse\\_www/wd\\_pds?p\\_web\\_site\\_id=1&p\\_web\\_page\\_id=5829&p\\_no\\_appla=Y&p\\_show=N](https://ert.cern.ch/browse_www/wd_pds?p_web_site_id=1&p_web_page_id=5829&p_no_appla=Y&p_show=N),
4. CERN, CMS Detector. 2008,  
<http://cms.web.cern.ch/cms/Detector/index.html>,2010
5. CERN, LHC Experiments. 2008,  
[http://lhc.web.cern.ch/lhc/lhc\\_experiments.htm](http://lhc.web.cern.ch/lhc/lhc_experiments.htm), 2010
6. CERN. The Large Hadron Collider. 2008, 2010.  
<http://public.web.cern.ch/public/en/lhc/lhc-en.html>
7. Collaboration C. *The CMS experiment at the CERN LHC*: CMS, CERN;2008.
8. Daems W. *Digital signal processing*. Hoboken: KDG.IWT; 2009.
9. DRAKA. *Coaxial cable specification*;2006.
10. Floyd T. *Digital fundamentals*. Upper Saddle River, N.J.: Pearson;2009.
11. Hunt BR, Lipsman RL, Rosenberg JM, e.a. *A guide to MATLAB: for beginners and experienced users*. 2 ed. ed. Cambridge: Cambridge University Press; 2002.
12. King RWP. *Transmission-line theory*. New York: Dover; 1965.
13. Marinela. CERENKOV EFFECT.  
[http://teachers.web.cern.ch/teachers/archiv/hst2002/bubblech/mbitu/cerenkov\\_effect.htm](http://teachers.web.cern.ch/teachers/archiv/hst2002/bubblech/mbitu/cerenkov_effect.htm).
14. P-N Designs I. A more exact coax attenuation solution. 2006;  
[http://www.microwaves101.com/encyclopedia/coax\\_exact.cfm](http://www.microwaves101.com/encyclopedia/coax_exact.cfm).
15. Arash P. C++ String Tokenizer.  
<http://www.partow.net/programming/stringtokenizer/index.html>.
16. Schildt H. *C++: the complete reference*. 4 ed. ed. New York: McGraw-Hill; 2003.
17. van, den Enden AWM, Verhoeckx NAM. *Digitale signaalbewerking*. Amerongen: Delta Press; 1994.

18. Wikipedia. Photomultiplier, <http://en.wikipedia.org/wiki/Photomultiplier>, 2010
  
19. X. Aslanoglou<sup>1</sup> AA, M.N. Bakirci<sup>2</sup>, S. Bas\_e\_ğmez<sup>2</sup>, W. Beaumont<sup>3</sup>, J. Blocki<sup>4</sup>, K. Borrás<sup>5</sup>, A. Campbell<sup>5</sup>, S. C\_erc\_i2 DdE, M. de Silva<sup>6</sup>, I. Dumano\_ğlu<sup>2</sup>, S. Erturk<sup>2</sup>, E. Eskut<sup>2</sup>, Y. Erchov<sup>7</sup>, P. G\_ottlicher<sup>5</sup>, L. Gouskos<sup>8</sup> GO, Y. Gusev<sup>9</sup>, H. Jung<sup>5</sup>, I. Katkov<sup>5,13</sup>, P. Katsas<sup>8</sup>, L. Khein<sup>5,13</sup>, F. Kisoglu<sup>2</sup>, A. Knutsson<sup>5</sup> SK, A. Kuznetsov<sup>7</sup>, M. Lebeau<sup>8</sup>, T. McCauley<sup>11</sup>, C. Muhl<sup>5</sup>, Y. Musienko<sup>11,12</sup>, S. Ochesanu<sup>3</sup> MO, K. Ozdemir<sup>2</sup>, S. Ozturk<sup>2</sup>, A.D. Panagiotou<sup>8</sup>, K. Shileev<sup>12</sup>, K. Sogut<sup>2,14</sup>, J. Swain<sup>11</sup>, S. Reucroft<sup>11</sup> MR, V. Tiflov<sup>12</sup>, A. Kayis-Topaksu<sup>2</sup>, H. Van Haevermaet<sup>3</sup>, P. Van Mechelen<sup>3</sup>. *Performance studies of the final prototype for the CASTOR forward calorimeter at the CMS experiment; 2009.*  
[http://cms.cern.ch/iCMS/jsp/openfile.jsp?type=NOTE&year=2008&files=NOTE2008\\_022.pdf](http://cms.cern.ch/iCMS/jsp/openfile.jsp?type=NOTE&year=2008&files=NOTE2008_022.pdf).
  
20. Xilinx, Virtex-II Platform FPGAs: Complete Data Sheet, 2007.  
<http://www.xilinx.com/support/documentation/virtex-ii.htm>
  
21. Xilinx, Virtex-II Platform FPGA User Guide, 2005,  
<http://www.xilinx.com/support/documentation/virtex-ii.htm>
  
22. Zimmerman Tom, Baumbaugh Alan HJ, Los Sergey, Shaw Theresa. *Specification for Production CMS QIE ASIC (QIE8)*: Fermilab;2002.

# APPENDIX

---

## APPENDIX I:LVDS

LVDS is een nieuwe technologie die gebruikt wordt voor high performance datatransmissies. Het maakt daarbij gebruik van differential data transmission waarover verder meer.

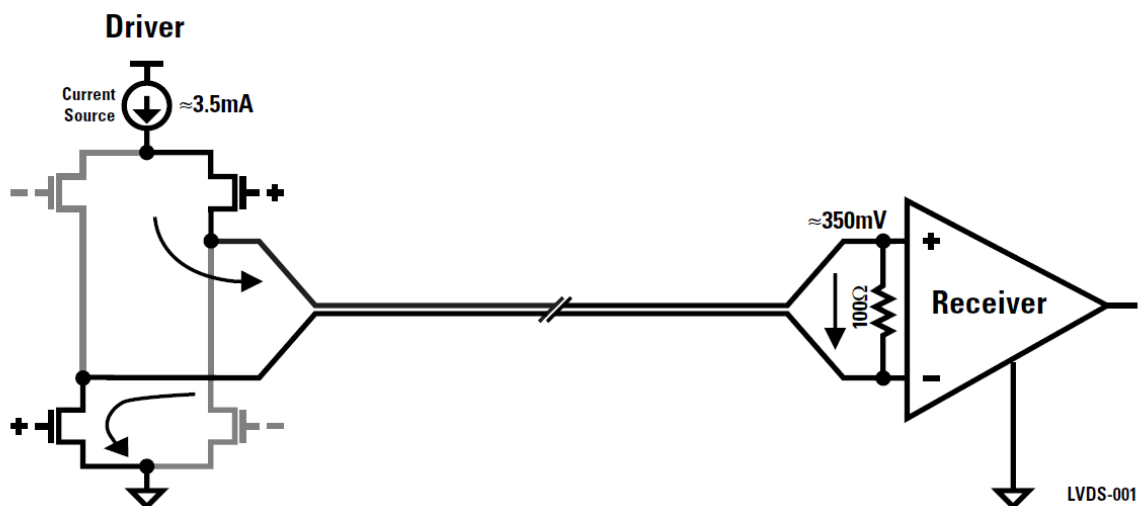
Er is meer en meer vraag naar hoge datarates over grote afstanden. LVDS laat toe om 100 Mbps tot 2 Gbps te versturen over het medium.

Enkele voordelen van LVDS:

- Low-voltage power supply
- Low-noise generaton
- High noise rejection
- Robust transmission signals(differential)
- Ability to be integreated into system level IC's!!

Hoe verkrijgen we snelheid zonder 'noise' en met een laag voltage?

LVDS maakt gebruik van een spanning van 350 mV. Deze lage spanning maakt het mogelijk snelle signaalveranderingen door te voeren (en dus hoge datarates te halen). Daarnaast zorgt deze lage spanning ook voor weinig dissipatie ( $P=UI$ ). Door gebruik te maken van differentiële technologie is er bijna geen invloed van storingen. LVDS maakt gebruik van twee lijnen met tegengestelde stroom en voltage. Wanneer een storing het medium beïnvloed, wordt de storing gekoppeld op de twee lijnen en blijft de nettoverschil 0.



Vermogengebruik:  $350\text{ mV} \times 3,5\text{mA} = 1,2\text{mW}$ .

Line termination



Figure 2.1. LVDS termination



Figure 2.1a. Internal termination

## Appendix II: Opstellen van de lijnvergelijking

$$(\alpha + \beta)^2 = (R + j\omega L)(G + j\omega C)$$

$$(\alpha^2 - \beta^2 + j2\alpha\beta = RG - \omega^2 LC + j\omega(LG + RC)$$

$$Re: \alpha^2 - \beta^2 = RG - \omega^2 LC$$

$$Im: j2\alpha\beta = \omega(LG + RC)$$

$$\alpha^2 = RG - \omega^2 LC - \beta^2$$

$$\beta = \frac{\omega(LG + RC)}{2\alpha} \implies \beta^2 = \frac{\omega^2(LG + RC)^2}{2\alpha^2}$$

$$\alpha^2 = (RG - \omega^2 LC) - \frac{\omega^2(LG + RC)^2}{2\alpha^2}$$

$$4\alpha^4 - 4\alpha^2(RG - \omega^2 LC) + \omega^2(LG + RC)^2 = 0$$

$$x = \alpha^2$$

$$4x^2 - 4(RG - \omega^2 LC)x + \omega^2(LG + RC)^2 = 0$$

$$V(x) = V_0 e^{-\gamma x}$$

$$V(x) = V_0 e^{-\left[\sqrt{\frac{1}{2}[(RG - \omega^2 LC) + \sqrt{(R^2 + \omega^2 L^2)(G^2 + \omega^2 C^2)}]} + j\sqrt{\frac{1}{2}[\sqrt{(R^2 + \omega^2 L^2)(G^2 + \omega^2 C^2)} - (RG - \omega^2 LC)]}\right]x}$$

$$\alpha = \sqrt{\frac{1}{2}\left((RG - \omega^2 LC) + \sqrt{(R^2 + \omega^2 L^2)(G^2 + \omega^2 C^2)}\right)} \text{ Dampingsconstante}$$

$$\beta = \sqrt{\frac{1}{2}\left(\sqrt{(R^2 - \omega^2 L^2)(G^2 + \omega^2 C^2)} - (RG - \omega^2 LC)\right)} \text{ Faseconstante}$$

Transferfunctie van de lijn

$$H(\omega, l) = V_0 \cos(\omega t - \alpha l) - j \sin(\omega t - \beta l)$$

$$H(\omega, l) = V_0 \cos\left(\omega t - \sqrt{\frac{1}{2}\left((RG - \omega^2 LC) + \sqrt{(R^2 + \omega^2 L^2)(G^2 + \omega^2 C^2)}\right)} l\right) - j \sin\left(\omega t - \sqrt{\frac{1}{2}\left(\sqrt{(R^2 - \omega^2 L^2)(G^2 + \omega^2 C^2)} - (RG - \omega^2 LC)\right)} l\right)$$

$$V(x) = V_0 e^{-(\alpha + j\beta)x}$$

### APPENDIX III: afleiding reflectiecoëfficiënt

$$\begin{aligned}\frac{V_i + V_r}{i_i + i_r} &= ZL \\ \frac{V_i + V_r}{\frac{V_i}{Z_0} + \frac{-V_r}{Z_0}} &= ZL \\ Z_0 \frac{V_i + V_r}{V_i - V_r} &= ZL \\ \frac{1 + \frac{V_r}{V_i}}{1 - \frac{V_r}{V_i}} &= \frac{ZL}{Z_0} \\ \frac{1 + \rho}{1 - \rho} &= \frac{ZL}{Z_0} \\ \rho &= \frac{ZL - Z_0}{ZL + Z_0}\end{aligned}$$

## APPENDIX IV:

Frequentie	-wT+phi	faseverschil
1	0	2,81337E-07
1,00E+03	0	0,000281337
1,00E+06	19	-18,71866334
1,00E+07	171	-168,1866334
2,00E+07	-20	-194,3732669
4,00E+07	-47	-215,7465338
6,00E+07	-67	-230,1198007
8,00E+07	-100	-257,4930676
1,00E+08	-123	-274,8663344
1,20E+08	-150	-296,2396013
1,30E+08	22	-345,4262348
1,40E+08	175	-495,6128682
1,50E+08	-14	-511,7995017
1,60E+08	145	-667,9861351
1,70E+08	-45	-745,1727686
1,80E+08	141	-928,359402
1,90E+08	-76	-1040,546035
2,00E+08	82	-1195,732669
2,10E+08	-111	-1401,919302
2,20E+08	53	-1563,105936
2,30E+08	-142	-1829,292569
2,40E+08	24	-1992,479203
2,50E+08	-176	-2321,665836
2,60E+08	-14	-2516,85247
2,70E+08	148	-2676,039103
2,80E+08	-52	-2757,225736
2,90E+08	104	-2910,41237
3,00E+08	-83	-3066,599003

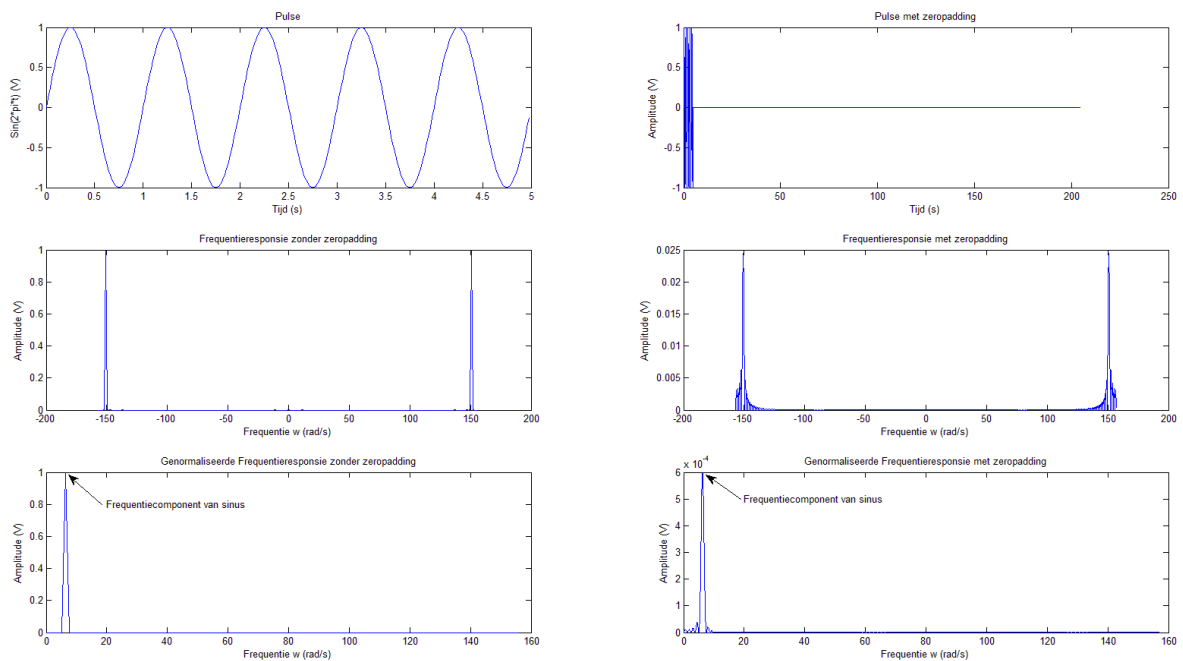
## APPENDIX V: Zero padding

Zeropadding is een techniek dat gebruikt kan worden om de resolutie in het tijdsdomein of het frequentiedomein te verhogen. Dit gebeurt door een reeks nullen toe te voegen aan de data. Door nullen toe te voegen, wordt het aantal samples verhoogt. Aan de hand van onderstaande formule kan vervolgens afgeleid worden dat op deze manier een grotere spectrale resolutie of een grotere tijdsresolutie bekomen kan worden.

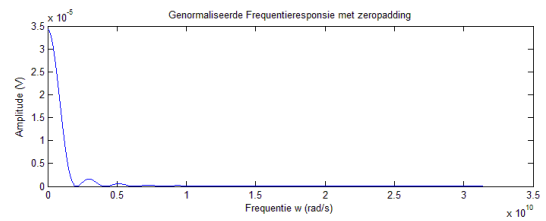
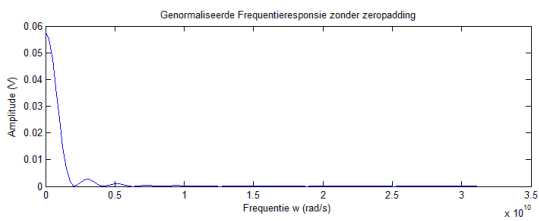
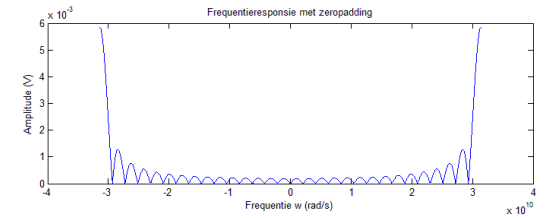
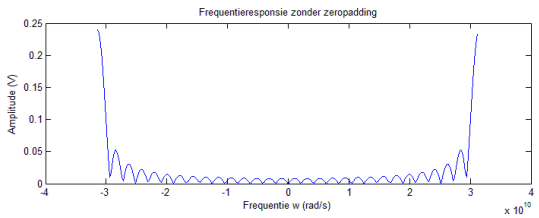
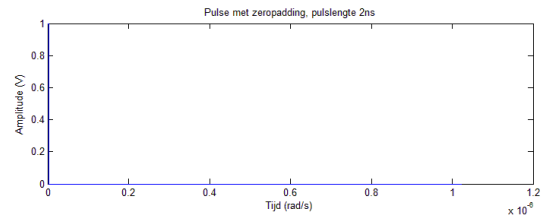
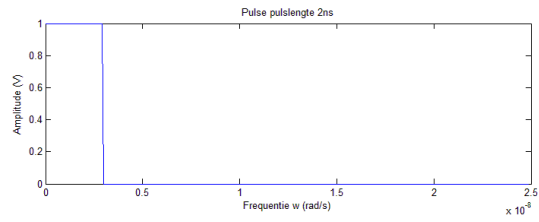
$$N = \frac{\omega_s}{\omega_f} \Leftrightarrow \omega_f = \frac{\omega_s}{N}$$

$$N = \frac{T_s}{T_f} \Leftrightarrow T_f = \frac{T_s}{N}$$

Op volgende figuur is een sinus getransformeerd naar het frequentiedomein. De linkse figuren zijn zonder zeropadding, de rechtse met zeropadding. Door zeropadding toe te passen, wordt de resolutie in het frequentiedomein vergroot.



Op volgende figuur is een puls van 6ns getransformeerd naar het frequentiedomein. Links zonder zeropadding, rechts met zeropadding.



Op volgende figuur is een puls van 2ns getransformeerd naar het frequentiedomein. Links zonder zeropadding, rechts met zeropadding.

