

# 3D Laserscanner

---

3D scannen van producten op CNC-freesbanken  
met behulp van lasertechniek

**Merijn ter Beek**

**30-5-2012**

## Samenvatting

Vanuit, met name de hobbymarkt, is er vraag naar het 3D-inscannen van producten in combinatie met CNC-freesbanken. Hiervoor bestaan dure complexe systemen die voor hobbyisten vanwege de prijs niet interessant zijn. Deze systemen komen vaak ook samen met ingewikkelde software waarbij na de meting nog veel moet gebeuren om een daadwerkelijk 3D-model te verkrijgen.

In dit rapport staat beschreven hoe een meetopstelling is gebouwd waarmee scans uitgevoerd kunnen worden met relatief goedkope componenten en een eenvoudige werking. Voor deze meetopstelling zijn componenten bij elkaar gezocht en is software geschreven om de meetopstelling te kunnen koppelen aan de huidige CNC-besturingen.

Met de meetopstelling is onderzocht of het principe werkt. Uit de tests is gebleken dat het principe inderdaad werkt. Er is met name gekeken naar de snelheid waarmee gemeten kan worden en de nauwkeurigheid ervan. Randeffecten zijn ook onderzocht en beschreven.

Voor de snelheid van het systeem bleek deze begrensd te worden door de besturing van de CNC-machine waarmee data van de meetopstelling werden ingelezen. Er kunnen elke 50 ms data worden ingelezen wat ervoor zorgt dat de snelheid waarmee scans uitgevoerd worden samenhangt met de gewenste nauwkeurigheid. Voor een nauwkeurigheid van 0,1 mm mogen er geen bewegingen met de machine gemaakt worden sneller dan 2,0 mm/s.

Een eis was dat de 3D-scans uitgevoerd konden worden met een nauwkeurigheid van 0,1 mm. De huidige opstelling voldoet hier niet aan. Er zijn vele factoren die aangepast kunnen worden om de nauwkeurigheid te verbeteren. De eenvoudigste manier is het gebruik van een krachtiger laser en met de uitlijning van de meetopstelling verbeteren.

<b>SAMENVATTING .....</b>	<b>2</b>
<b>HOOFDSTUK 1: INLEIDING .....</b>	<b>4</b>
<b>HOOFDSTUK 2: THEORIE.....</b>	<b>5</b>
VERMOGEN VAN LICHT.....	5
LAMBERTSE-STRALER.....	5
ABSORPTIE VAN LICHT.....	6
FOTO-DIODE .....	6
LASER-DIODE .....	6
OP-AMP .....	7
RUIS.....	9
<b>HOOFDSTUK 3: MEETOPSTELLING HARDWARE .....</b>	<b>10</b>
INTRODUCTIE MEETOPSTELLING .....	10
HET IDEE ACHTER DE MEETOPSTELLING.....	11
OPBOUW .....	12
COMPONENTEN.....	14
<b>HOOFDSTUK 4: MEETOPSTELLING SOFTWARE.....</b>	<b>17</b>
<b>HOOFDSTUK 5: METINGEN .....</b>	<b>22</b>
STABILITEIT VAN DE LASER .....	22
RUIS.....	24
SNELHEID VAN HET SYSTEEM .....	24
VERTRAGING.....	29
METING MET DE XY-SCANNER .....	31
INMETEN VAN HOOGTE .....	35
<b>HOOFDSTUK 6: CONCLUSIE .....</b>	<b>40</b>
<b>BIBLIOGRAFIE.....</b>	<b>41</b>
<b>BIJLAGE .....</b>	<b>42</b>
A. SOURCE CODE VAN DE MAINWINDOW.....	43
B. SOURCECODE VAN CALIBRATIE METING.....	50
C. SOURCECODE VAN XY-SCAN.....	52
D. REKENMODEL .....	55
E. MEETRESULTATEN VAN LASERDIODE ADL-65075TR/L.....	58
F. DATASHEETS .....	59

## Hoofdstuk 1: Inleiding

DamenCNC is een jong bedrijf dat actief is in de markt op het gebied van CNC frees- en draaibanken. Het begon met het leveren van CNC besturingen voor freesbanken maar groeit steeds meer uit tot een bedrijf dat alles kan leveren op het gebied van CNC machines. Klanten die een besturing hebben aangeschaft kwamen regelmatig met de vraag of het mogelijk was om met hun machine een product in te kunnen meten.

Om een product in te meten kan gebruik worden gemaakt van een touchprobe. Deze kan het product stap voor stap aftasten. Het is een probe die mechanisch reageert wanneer er een kleine kracht op wordt uitgeoefend. De probe beweegt naar een coördinaat toe van de machine en kan registreren wanneer deze probe geactiveerd wordt. De touchprobe heeft een aantal gebreken:

- Metingen verlopen vrij traag omdat het op aanraking reageert.
- Meting is niet zuiver genoeg, de richting waarin de kracht werkt op de probe is bepalend voor de tijd dat deze reageert.
- De touchprobe is onbetrouwbaar, hij werkt met mechanische componenten die niet altijd even betrouwbaar zijn.
- Erg lastig produceerbaar, lange assemblage tijd en dus ook kostbaar.

Het doel van het onderzoek is het presenteren van een nieuw ontwerp voor een meetinstrument met behulp van een laser. Er wordt besproken hoe de opstelling tot stand is gekomen en hoe deze werkt. Resultaten die behaald zijn met het meetinstrument zullen ook besproken worden, hierbij speelt vooral de nauwkeurigheid een belangrijke rol.

In hoofdstuk 2 staat een deel beschreven van de onderliggende theorie. Hoofdstuk 3 beschrijft de meetopstelling met aanvullende theorie. In hoofdstuk 4 staat beschreven over hoe de software is opgebouwd. In hoofdstuk 5 staan meerdere metingen die verricht zijn met een zelfgebouwde meetopstelling. De conclusies zijn terug te vinden in hoofdstuk 6.

## Hoofdstuk 2: Theorie

Om met een laser metingen te gaan verrichten is kennis nodig op enkele natuurkundige vlakken. In dit hoofdstuk zullen enkele formules met betrekking tot licht genoemd worden en zal er beknopte theorie gegeven worden over componenten die gebruikt zijn bij de meetopstelling.

### *Vermogen van licht*

Licht kan op meerdere manieren worden beschreven, voor dit rapport is licht het beste te beschrijven met fotonen. Een foton bestaat uit energie en heeft geen rustmassa. Een lichtstraal is een bundel van fotonen waarbij het vermogen van alle fotonen tezamen het vermogen van de lichtstraal oplevert.

Een gangbare manier om het vermogen van licht uit te drukken is de stralingsfluxdichtheid. Deze is te berekenen met formule [1].

$$\varphi_e = \frac{\Phi_e}{A} \quad [1]$$

met:

$\varphi_e = \text{stralingsfluxdichtheid } [W/m^2]$

$\Phi_e = \text{stralingsflux } [W]$

$A = \text{Oppervlakte } [m^2]$

### *Lambertse-straler*

Bij een lambertse-straler geldt dat een invallende lichtstraal in alle richtingen evenveel wordt verstrooid. In feite is een lambertse-straler het tegenovergestelde van een spiegeland oppervlak, waarbij licht slechts in één richting wordt gereflecteerd. Dit betekent dat een puntbron met een bepaald vermogen gelijkmatig als een halve bol zal worden verdeeld in de verstrooide straling. De stralingsfluxdichtheid is nu omgekeerd evenredig met de oppervlakte van de halve bol.

$$\varphi_e = \frac{\Phi_e}{4\pi r^2} \quad [2]$$

met:

$r = \text{afstand tot de bron } [m]$

Wanneer er nu een deel van de oppervlakte op de halve bol genomen wordt, is te berekenen met de stralingsfluxdichtheid hoeveel vermogen er op dat oppervlak valt.

Formule [2] geldt alleen bij een ideale lambertse-straler. Bij een niet-ideale lambertse straler is het lastiger te voorspellen hoeveel vermogen er op een bepaald oppervlakte valt van de halve bol. In bepaalde richtingen zal meer of minder licht worden uitgezonden dan in andere richtingen. De mate

van lambertse straling is afhankelijk van het type materiaal en de bewerking die daarop heeft plaatsgevonden.

### Absorptie van licht

Fotonen kunnen van een materiaal weerkaatst worden, maar ze kunnen ook geabsorbeerd worden door het materiaal. De energie van de foton kan worden opgenomen door elektronen in het materiaal en wordt dan omgezet in bijvoorbeeld warmte. De foton zal dus niet weerkaatsen in de vorm van licht. De mate van absorptie is afhankelijk van de golflengte van het licht en het materiaal. Materialen met een lichte kleur absorberen weinig fotonen terwijl materialen met een donkere kleur bijna alle fotonen absorberen.

### Foto-diode

Een foto-diode bestaat uit twee halfgeleiders waarvan de één van het n-type is en de ander van het p-type. Samen vormen ze een zogenaamde pn-junctie waarbij invallende fotonen elektronen vrijmaken. De vrijgemaakte elektronen zorgen ervoor dat een stroom gaat lopen door de fotodiode.



Figuur 1 Een voorbeeld van een fotodiode in een behuizing met lenswerking. Bron: Farnell.com

De stroom die ontstaat in een foto-diode is afhankelijk van het aantal fotonen (dus het vermogen) dat op de fotodiode valt en de  $A/W$  waarde van de fotodiode.

$$I_{foto} = P_{foto} \cdot A/W_{foto} \quad [3]$$

met:

$I_{foto}$  = stroom die gaat lopen door de foto – diode [A]

$P_{foto}$  = vermogen dat wordt opgenomen door de foto – diode [W]

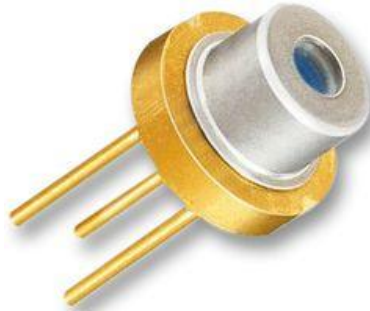
$A/W_{foto}$  = AW waarde van de foto diode geeft een ratio aan van de hoeveelheid stroom die wordt opgewekt bij een invallend vermogen [A/W]

De materiaalkeuze van de halfgeleiders is bepalend voor de waarde van  $A/W_{foto}$  en de gevoeligheid voor bepaalde golflengtes.

### Laser-diode

De laserdiode bestaat in zijn simpelste vorm uit een pn-junctie die wordt gepompt door een elektrische stroom. De stroom zorgt ervoor dat elektronen er in aangeslagen worden en wanneer ze

terug vallen naar hun originele positie zullen ze fotonen uitzenden. Door een slimme keuze van het halfgeleider materiaal kan de halfgeleider zelf een trilhaar vormen voor de laser. Dit betekent dat er minimale onderdelen nodig zijn voor dit type laser en dat deze klein kan worden gehouden. Deze factoren houden de kosten laag en maakt de laserdiode een zeer efficiënte, kleine en relatief goedkope laser.



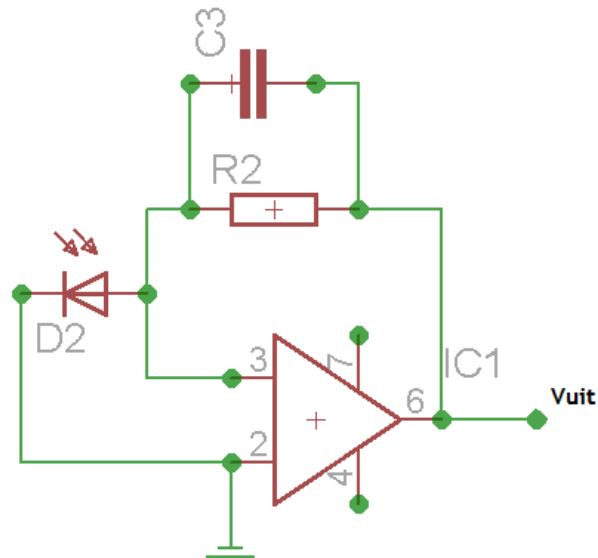
Figuur 2 Een typisch voorbeeld van een laserdiode. Bron: Farnell.com

Uitgangsvermogen van een laser zit typisch in de enkele mW's maar kan oplopen tot enkele honderden mW's. De stroom die nodig is voor de laserwerking varieert dan ook tussen de tientallen milliampères tot enkele honderden milliampères. Voordeel van de lage uitgangsvermogen lasers is dat ze op batterijen kunnen werken of kleine voedingsbronnen. Een diode laser zendt (afhankelijk van de temperatuur) een laserbundel uit met één specifieke golflengte.

Een laserdiode zendt een elliptische laserbundel uit. Voor vele toepassingen is een elliptische laserbundel niet gewenst dus daarom is er vrijwel altijd ook een lens nodig voor de laserdiode. Veel diodelasers worden standaard uitgerust in een behuizing met een ingebouwde lens.

### OP-AMP

De stroom die gaat lopen door de foto-diode ten gevolge van een invallend vermogen is typisch in de ordegrootte van enkele microampères. Om een dergelijke stroom toch meetbaar te maken wordt deze versterkt met een OP-AMP. Een veel gebruikte configuratie voor het versterken van een foto-diode signaal is weergegeven in Figuur 3. De OP-AMP functioneert hier als een transimpedantie versterker.



Figuur 3 Schematische weergave van het op-amp circuit waar de fotodiode in geschakeld is.

De componenten van Figuur 3 zijn als volgt:

$C3 = C_f$

$R2 = R_f$

D2 = Foto-diode

IC1 = OP-AMP met:

2	- ingang
3	+ ingang
7	OP-AMP voeding +
4	OP-AMP voeding -
6	Uitgang

Bij deze configuratie geldt voor de  $V_{uit}$ , de uitgangsspanning, dat deze kan worden bepaald met:

$$V_{uit} = I_{foto} R_f \quad [4]$$

met:

$V_{uit}$  = spanning aan de uitgang van op – amp [V]

$R_f$  = Weerstand die parallel staat over de uitgang en ingang van de op – amp [Ohm]

De combinatie van  $C_f$  en  $R_f$  zorgt er ook voor dat er sprake is van een RC-tijd, ook wel  $\tau$ -tijd genoemd. Dit is de tijd die nodig is om de condensator tot 63% van zijn totale eindvermogen op te laden. De  $\tau$  - tijd kan berekend worden met formule [5].

$$\tau = R_f C_f \quad [5]$$



## *Ruis*

### **Hagelruis en generatie en recombinatie ruis (G/R)**

Beide ruisvormen komen vooral veel in de optoelektronica voor omdat ze typerend zijn bij pn-overgangen. Typische componenten met pn-overgangen zoals laserdiodes en fotodiodes hebben dus last van deze ruisvormen. De processen in de pn-overgang waarbij fotonen en elektronen een rol spelen gebeuren stochastisch, dat wil zeggen dat de processen willekeurig plaatsvinden. Dit zorgt ervoor dat er een fluctuatie ontstaat in overgangen van fotonen naar elektronen.

### **Thermische ruis**

Alles wat stroom geleidt met een temperatuur groter dan het absolute nulpunt veroorzaakt ruis. Wanneer de geleider een temperatuur heeft hoger dan het absolute nulpunt zullen de elektronen binnen in de geleider zich chaotisch gaan bewegen wat de ruis veroorzaakt.

## Hoofdstuk 3: Meetopstelling hardware

In dit hoofdstuk komt aan bod hoe de meetopstelling tot stand is gekomen en uit welke onderdelen deze is opgebouwd en waarom. Allereerst zal er een korte introductie gegeven worden over bestaande opstellingen en hoe de huidige opstelling tot stand is gekomen. Als volgt zal op componenten niveau uitleg worden gegeven over de opstelling.

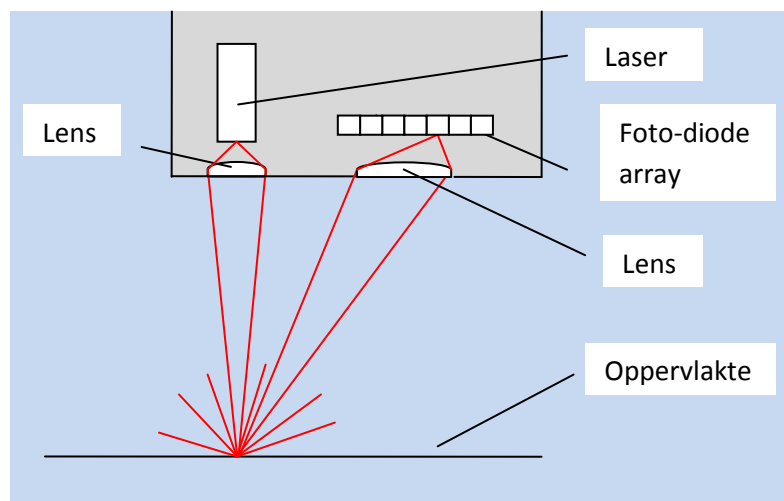
### *Introductie meetopstelling*

Op de CNC-markt bestaan al meerdere systemen die met behulp van lasers 3D-scans van producten kunnen maken. Deze systemen zijn echter enorm kostbaar en werken met dure softwarepakketten. De metingen worden vaak snel verricht, maar daarna zit er nog veel tijd in om instellingen te doen om uiteindelijk een 3D-model te krijgen van het ingescande product.

De nauwkeurigheid van de huidige systemen laat vaak ook nog te wensen over, ze halen in de meeste gevallen niet de beloofde nauwkeurigheid.

Met name voor de hobbymarkt zou een betaalbare scanner die eenvoudig te bedienen is een uitkomst zijn. Er is dus gefocust om een relatief goedkope scanner te maken die eenvoudig is in het gebruik.

Voor het inmeten van producten moeten er afstanden gemeten worden. Een veelgebruikte techniek hiervoor is het triangulatie principe. Hierbij wordt gemeten onder welke hoek een lichtstraal de detector binnenvalt om zo terug te rekenen naar een afstand. Figuur 4 laat zien hoe dit uitgevoerd kan worden.



**Figuur 4** Schematische weergave van het triangulatie principe. Afhankelijk van de afstand focuseert de lens de lichtstralen op een andere plaats van de fotodiode array. Zo kan de hoek worden bepaald.

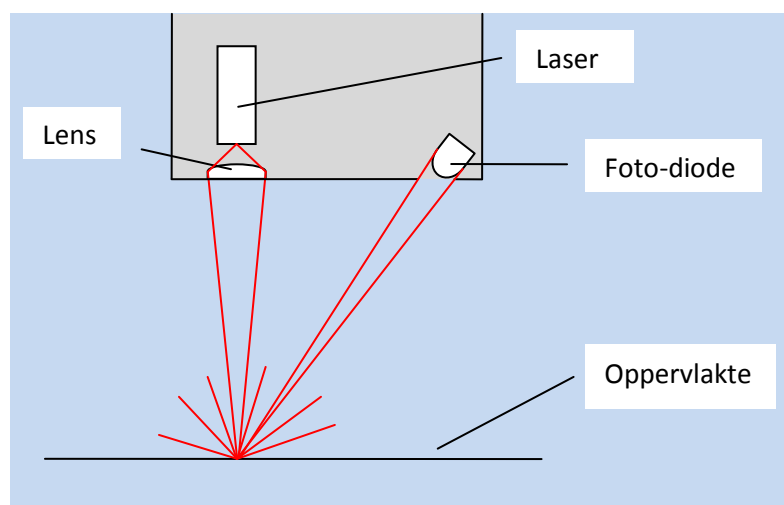
Wanneer een dergelijk systeem zich over een product heen beweegt kan het op elk punt de hoogte van het product meten. Op deze manier kan er dus een hoogtekaart gemaakt worden van het

product waardoor er een 3D model ontstaat. Dit principe kan nog verder worden uitgebreid wanneer er gebruik wordt gemaakt van een lijnlaser en een CCD-chip. Op die manier kunnen tegelijkertijd alle hoogten die vallen op de geprojecteerde lijn bepaald worden.

Dergelijke systemen bevatten echter dure componenten en vereisen een grote precisie bij montage en afstelling. Dit maakt het voor kleinschalige productie erg kostbaar en bovendien zijn ze lang niet altijd even nauwkeurig.

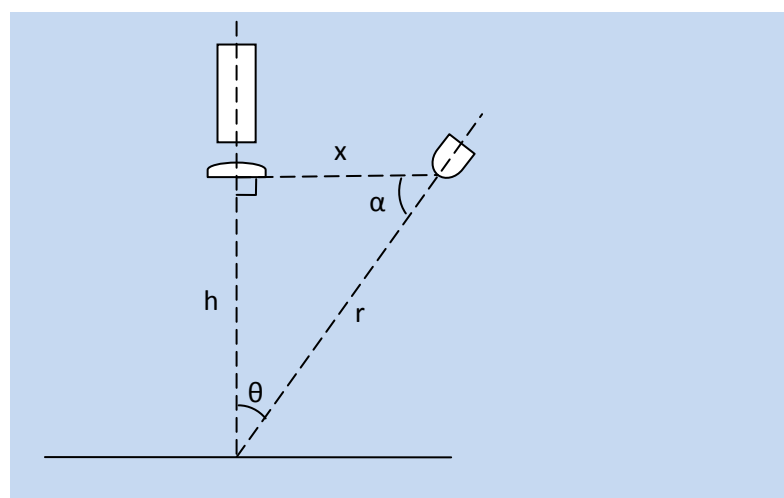
### *Het idee achter de nieuwe meetopstelling*

Omdat het meetinstrument uiteindelijk aan een CNC-machine wordt gekoppeld kan dat feit gebruikt worden voor een goedkopere en veel makkelijker af te stellen meetinstrument. De foto-diode array en de lens uit Figuur 4 worden vervangen voor één eenvoudige foto-diode. Dit levert het resultaat op te zien in Figuur 5.



**Figuur 5** Schematische weergave van hoe het licht van de laser terecht komt bij de detector.

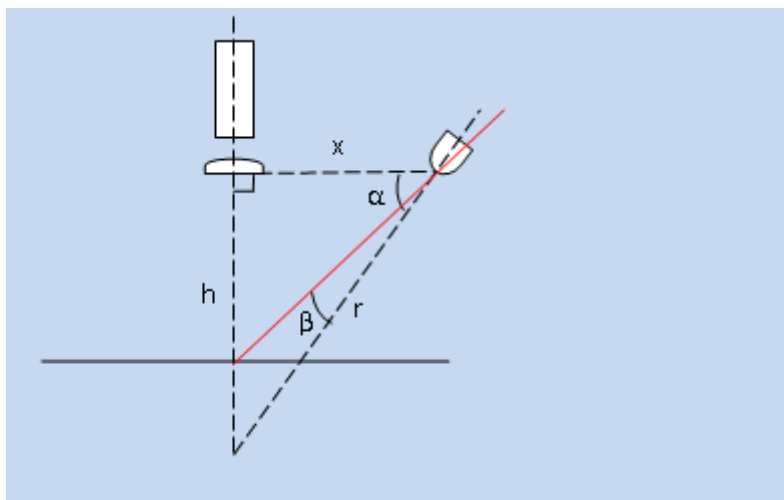
De fotodiode is nu onder een vaste hoek geplaatst. Dit houdt in dat hij reageert bij een vooraf ingestelde hoogte. Figuur 6 en Figuur 7 illustreren het principe hiervan.



**Figuur 6** De denkbeeldige driehoek met belangrijke afstanden en hoeken die wordt gebruikt bij de berekeningen.

In Figuur 6 is te zien dat  $x$  en  $\alpha$  in een vaste positie zijn gezet. De hoogte  $h$  kan worden gevarieerd door middel van een CNC-besturing. Het opgenomen vermogen van de fotodiode is in dit geval dus afhankelijk van de afstand  $r$  tot de bron en de hoek  $\alpha$  waarbij het vermogen binnenvalt. Door een slimme keuze van de fotodiode kan deze gevoelig gemaakt worden voor de hoek waarin het licht binnenkomt. De stroom die opgewekt wordt in de fotodiode zal dus afhangen van de straal  $r$  en de hoek  $\alpha$ . Zowel  $r$  en  $\alpha$  zijn weer afhankelijk van de variabele  $h$ . Wanneer  $h$  dus varieert zal er een karakteristiek ontstaan waarbij er een piek waar te nemen is in de hoeveelheid opgewekte stroom bij de ideale  $h$ .

Wanneer de hoogte  $h$  kleiner wordt gemaakt dan in Figuur 6 ontstaat de situatie geschetst in Figuur 7. De lichtstraal komt onder een andere hoek de foto-diode binnen waardoor, bij hetzelfde vermogen van de laser, er een ander vermogen wordt gedetecteerd door de fotodiode.

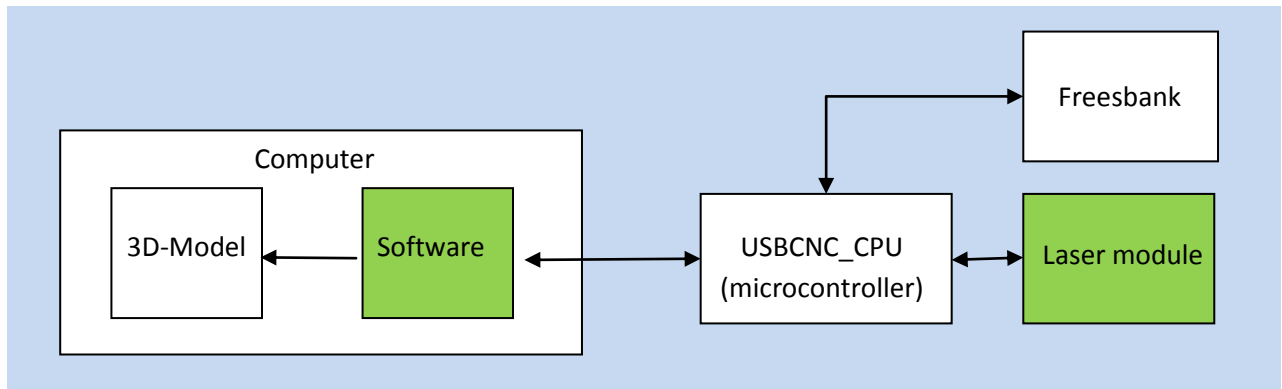


**Figuur 7** Illustratie van wat er gebeurt wanneer het meetinstrument zich dicht bij het oppervlak bevindt. Bij het maximaal opgenomen vermogen is hoek  $\beta$  gelijk aan 0. In dit geval is  $\beta$  groter dan 0 waardoor de gevoeligheid van de fotodiode veranderd.

De piek in het vermogen vindt altijd plaats bij dezelfde  $h$ , ongeacht het materiaal waarop de lichtstraal weerkaatst, zelfs wanneer het bijvoorbeeld geen lambertst-straler is. Wel zal de signaal/ruis verhouding slechter worden bij niet lambertse-stralers.

### *Opbouw*

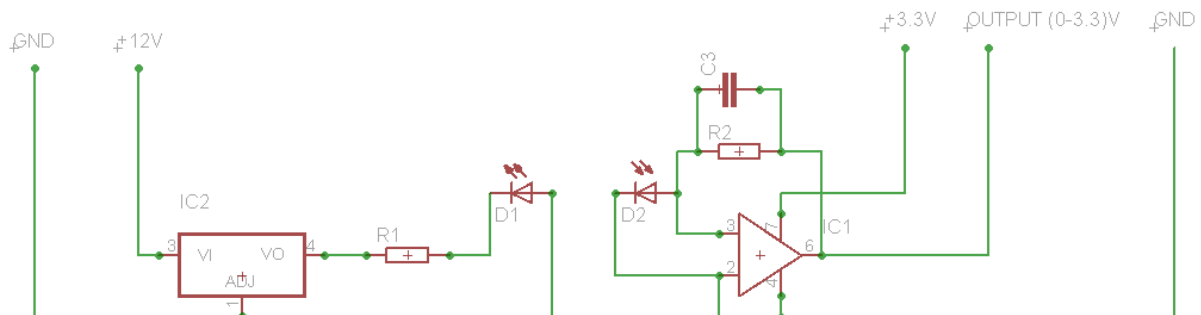
De meetopstelling moet gekoppeld worden aan een computer om de data te verwerken. Hoe deze structuur is opgebouwd is te zien in het volgende blokschema.



**Figuur 8** Blokschema dat de volgorde van het proces aangeeft.

Het blokschema van Figuur 8 geeft weer hoe het meetsysteem werkt. Een computer, waarop uiteindelijk de gegevens moeten komen is gekoppeld aan een microcontroller die normaal gesproken de besturing regelt van de freesbank. De microcontroller stuurt en ontvangt informatie van de freesbank. De lasermodule zend en ontvangt ook informatie van de microcontroller. Deze stuurt de informatie dan uiteindelijk weer door naar de computer. De informatie van de lasermodule en freesbank worden aan elkaar gekoppeld door de software op de computer.

De lasermodule zelf is opgebouwd uit de onderdelen die schematische zijn weergegeven in Figuur 9.



**Figuur 9** Schematische weergave van opbouw van de meetopstelling.

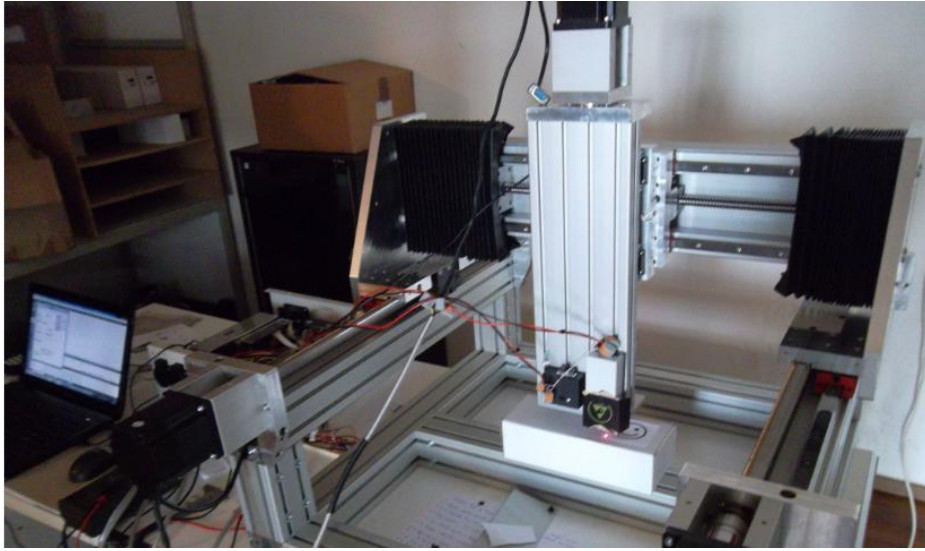
De laser D1 wordt aangedreven door een stroomregelaar die de stroom door de fotodiode constant houdt. De stroomregelaar wordt gevoed met 12V, afkomstig van USBCNC CPU. De fotodiode D2 is gekoppeld aan een OP-AMP.

De microcontroller beschikt standaard over verschillende in- en uitgangen. Hiervan wordt 1 uitgangssignaal gebruikt om de laser aan en uit te kunnen zetten en 1 ingang wordt gebruikt om de foto-diode uit te kunnen lezen. De informatie van de fotodiode komt bij de microcontroller binnen op een analoge ingang. Deze ingang zit direct aangesloten op de processor en mag een maximale spanning van 3,3V krijgen. Om te voorkomen dat dit nooit wordt overschreden wordt de voedingspanning van de op-amp op 3,3V vastgesteld. De 3,3V wordt eveneens direct afgetapt van de microcontroller.

## Componenten

### Freesbank

De freesbank die gebruikt is bij de metingen is een door DamenCNC zelf ontwikkelde machine die werkt op een DamenCNC RTR Performance besturing. De machine is opgebouwd met een goede stijfheid. De freesmachine kan ruwweg een nauwkeurigheid van 0,01 mm halen. Dit is een factor 10 beter dan de 0,1 mm die gemeten wenst te worden met de laser scanner.



**Figuur 10** De freesbank waarop de metingen zijn verricht.

### Laserdiode

De laserdiode die gebruikt is voor de metingen beschreven in dit rapport is de PLP6501AR. De golflengte van deze laserdiode is 650 nm en heeft een maximaal uitgangsvermogen van 0,99  $\mu$ W. De golflengte keuze is belangrijk bij de laserdiode, deze moet ook in overeenstemming zijn met de gebruikte fotodiode. Een goedkoop materiaal voor de pn-junctie wat veel voor laserdiodes wordt gebruikt geeft de laserdiode een golflengte van rond de 650 nm, om die reden is dit een relatief goedkoop model. Het vermogen van de laser moet zo hoog mogelijk zijn voor een zo nauwkeurig mogelijke meting maar om kosten laag te houden en het product veilig te houden is een lager uitgangsvermogen gewenst.

De stroom die door de laserdiode loopt wordt geregeld met een spanningsregelaar die als stroombron is opgesteld. De stroom die gaat lopen door de laserdiode wordt constant gehouden op 27mA. Dit zorgt voor een stabiel uitgangsvermogen van rond de 0,6 mW.

De laserdiode zit ingebouwd in een behuizing die fungeert als een koellichaam en tevens in staat is om de ingebouwde lens te kunnen afstellen op een bepaald brandpunt. Het brandpunt kan worden ingesteld van vanaf 6cm.

## USBCNC CPU

De USBCNC CPU werkt met een PIC32MX460F microcontroller. Van deze microcontroller is de analoge ingang van belang. Dit is een 10-bits AD-converter die 1 miljoen samples per seconde verwerkt. Het spanningsbereik van de AD-converter is 0V–3,3V. Dit betekent dat de LSB (least significant bit) waarde van de AD-converter 0,032mV bedraagt.

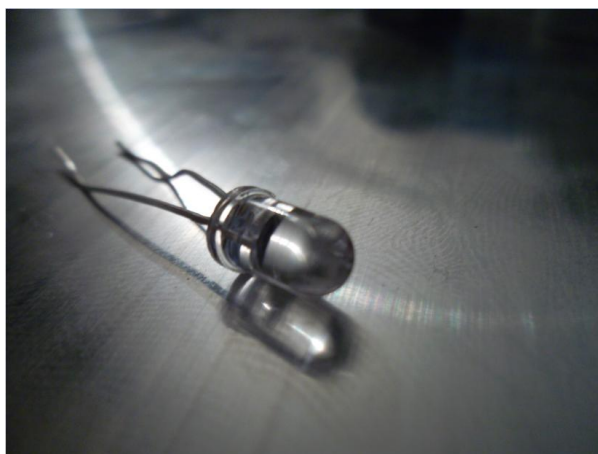
Om de laserdiode aan of uit te zetten wordt er gebruik gemaakt van een van de standaard uitgangen van USBCNC CPU. De uitgangen zijn van het type NPN en schakelen dus naar de GND. De USBCNC CPU beschikt over een externe voeding die kan worden aangebracht tot 24V. Deze externe voeding wordt doorverbonden met de driver voor de laserdiode (welke minimaal 12V nodig heeft). Wanneer de uitgang wordt aangestuurd schakelt deze naar de GND waardoor de driver de benodigde spanning krijgt om de laserdiode aan te sturen.

## Fotodiode

De gebruikte fotodiode is de OSRAM SFH 213. Een aantal aspecten waren belangrijk voor het selecteren van een geschikte fotodiode:

- Golflengte detectie, de fotodiode moet gevoelig zijn voor de golflengte die de laserdiode uitstraalt. In het optimale geval reageert hij niet op afwijkende golflengtes.
- Gevoeligheid, zo hoog mogelijke gevoeligheid aangezien het hier om een relatief zwak te meten signaal gaat.
- De donkerstroom dient zo klein mogelijk te blijven om de signaal-ruis verhouding zo klein mogelijk te houden.
- Halve-hoek, de halve-hoek van de fotodiode dient zo klein mogelijk te zijn om zo groot mogelijke piek te krijgen.

De tijd die nodig is om de fotostroom te laten lopen is ook van belang voor de snelheid van het systeem. Bij fotodiodes duurt het typisch enkele nanoseconde voordat de stroom is bereikt die hoort bij een bepaald opvallend vermogen.



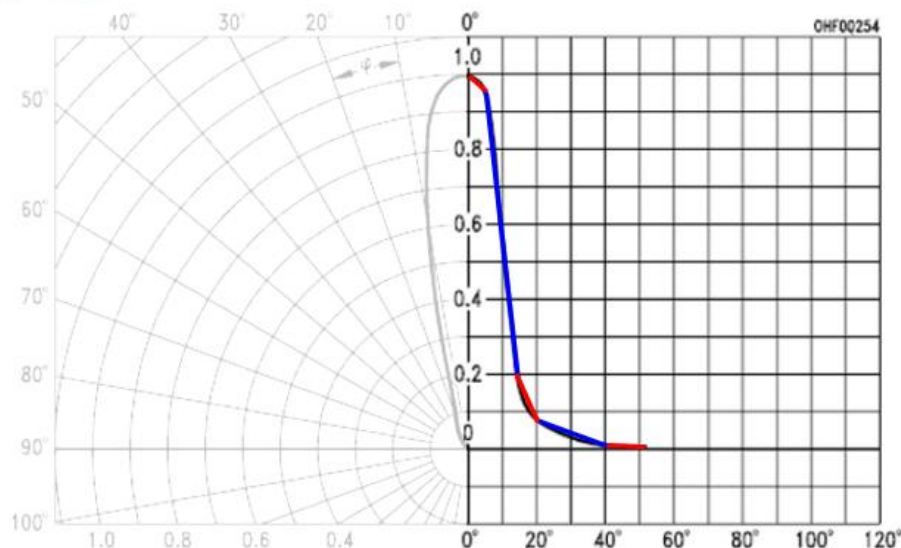
**Figuur 11 De OSRAM SFH 213**

De OSRAM SFH 213 zit in een T-1 3/4 behuizing. Dit is gunstig voor de detectiehoek. De behuizing is in feite een van zichzelf een lens waardoor de zogenaamde halve hoek zo klein mogelijk wordt

gehouden. Wanneer de halve hoek klein is, zal het verschil in een kleine hoekvariatie groter worden. De halve hoek van dit type foto-diode is 10°.

### Directional Characteristics

$$S_{rel} = f(\varphi)$$



Om in een rekenmodel de berekeningen automatisch te doen moet de waarde van  $S_{rel}$  ook bepaald worden voor meerdere hoeken. Er is geen functieomschrijving aanwezig van deze curve. De curve is daarom opgesplitst in 5 rechte lijnen. Hiervan zijn de begin en eindpunten bepaald en is er zo een richtingscoëfficiënt berekend voor elke lijn. Het model zoekt vanzelf de juiste lijn bij de hoek en berekent vervolgens de plaats op de lijn met de bijbehorende  $S_{rel}$ . Het gebruikte rekenmodel om  $S_{rel}$  te bepalen is terug te vinden in Bijlage D in de source code van het rekenmodel onder de functie GetAW().

### Op-Amp

De stroom die wordt opgewekt in de fotodiode is in de orde van de  $\mu A$ . Dit signaal moet worden versterkt met een Op-Amp waarbij zo min mogelijk ruis ontstaat. De Op-Amp die gebruikt is voor de metingen beschreven in dit rapport is de MCP6024. Dit is een IC met daarin 4 OP-AMPS. Gedetailleerde informatie over de Op-Amp is te vinden in de datasheet van Bijlage F.

Buiten de op-amp is de condensator keuze die deze aanvult ook van belang. Een ideale op-amp zou instantaan reageren wanneer er een signaal wordt aangeboden en deze in de juiste maten versterken en een stabiel uitgangssignaal geven. In werkelijkheid heeft de OP-AMP tijd nodig om te reageren, schiet het signaal vaak uit en is er ruis aanwezig.



## Hoofdstuk 4: Meetopstelling Software

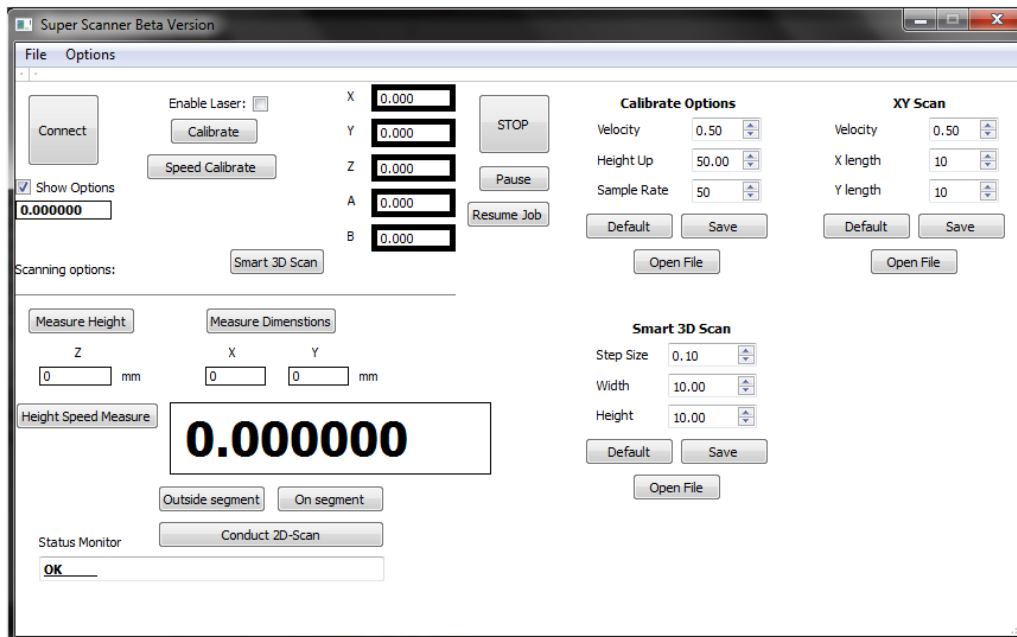
In dit hoofdstuk staat beschreven met welke software er gewerkt is om de meetopstelling aan te sturen. Hierbij zijn ook 2 voorbeelden gegeven van programma's die geschreven zijn en die veel zijn gebruikt voor het uitvoeren van metingen.

De USBCNC-CPU komt samen met het USBCNC-Software pakket. Dit is een speciaal ontwikkeld software pakket gericht op het frezen en draaien. De software communiceert door middel van een API<sup>[\*]</sup> met de USBCNC CPU. Sinds kort is het mogelijk om ook als klant zelf software te schrijven voor de USBCNC CPU. De API kan dan gebruikt worden om te communiceren met de CPU en in een software omgeving, in dit geval C++, kunnen er dan programma's geschreven worden. Deze optie is pas net beschikbaar en daardoor was er ook nog geen ervaring binnen DamenCNC met het programmeren van die combinatie. Het bevindt zich ook nog in een vroeg stadium waardoor er geen uitgebreide handleidingen beschikbaar zijn.

Om de software te schrijven moest er nog veel onderzoek worden gedaan in hoe de API van USBCNC werkt en hoe deze gekoppeld kon worden met C++. Voor het programmeren is gebruik gemaakt van Qt-Creator. Dit is een uitgebreid programma op C++ basis waarmee het mogelijk is om GUI's<sup>[\*\*]</sup> te programmeren

[\*] API staat voor Application Programming Interface. Het kan worden gezien als een soort tussenstap in communicatie tussen twee programma's. Twee verschillende programma's kunnen met elkaar communiceren door middel van een API waardoor beide programma's alleen hoeven te weten hoe de API werkt en hoeven ze niet van elkaar te weten hoe ze werken. De meeste printers worden bijvoorbeeld aangestuurd door een API, een tekstverwerker hoeft niet te weten hoe de printer werkt dus stuurt deze informatie naar een API die vervolgens het printwerk met de printer regelt.

[\*\*] GUI staat voor Graphical User Interface. Dit zorgt voor een gebruiksvriendelijke manier om een programma te gebruiken. Het visualiseert bijvoorbeeld acties die kunnen worden ondernomen met knoppen die kunnen worden ingedrukt. Een voorbeeld van een GUI is bijvoorbeeld een tekstverwerker waarbij je niet hoeft te programmeren om bijvoorbeeld een lettertype te kunnen veranderen maar deze simpelweg kunt aanklikken in de GUI.



Figuur 12 Screenshot van de zelfgemaakte GUI die communiceert met de USBCNC CPU en met de huidige software van USBCNC.

De software is zodanig geschreven dat het in ieder geval metingen kan automatiseren en versnellen maar tegelijkertijd is het zo opgebouwd dat het ook gebruikt kan worden voor een eindproduct. Het programma is opgebouwd uit verschillende threads elke thread kan worden opgeroepen vanuit het hoofdscherm en voert dan een actie uit. Dit maakt het overzichtelijk maar is ook noodzakelijk. Een thread zorgt ervoor dat er een apart proces wordt aangemaakt om de actie uit te voeren. Wanneer dit niet zou gebeuren zou de hoofd GUI vastlopen bij acties die veel eisen van de processor.

De sourcecode van de Mainwindow is te vinden in de Bijlage A. Vanuit de Mainwindow wordt alles geregeld. Hier is waar het programma start, de Mainwindow is opgebouwd samen met de Mainwindow.ui. De extensie .ui komt van User Interface, dit is een speciale sectie bij Qt-creator waar de GUI vorm kan worden gegeven. Knoppen (pushbuttons) die geplaatst worden kunnen door middel van functies in de mainwindow aangestuurd worden. In dit geval worden deze vooral gebruikt om een thread op te roepen die dan een actie gaat uitvoeren.

De API van USBCNC bestaat in feite uit functies die kunnen worden aangeroepen. Deze functies zijn dus verwerkt in de zelfgeschreven Mainwindow en threads. Een voorbeeld van een functie is bijv.:

```
CncMoveTo(CNC_POSITION pos,int move[],double velocityFactor);
```

Door deze functie op te roepen kan de machine worden aangestuurd dat hij naar een bepaalde positie beweegt. De positie wordt aangegeven met een struct waarin 6 coördinaten staan, welke assen daadwerkelijk moeten gaan bewegen kan worden aangegeven met een array die per as aangeeft wel (1) of niet (0) te bewegen. Vervolgens kan de snelheid waarmee dit alles moet gebeuren ook worden meegegeven als de velocityFactor.

De twee threads die het meest gebruikt zijn om metingen te verrichten zullen nu nader worden toegelicht:

### **Calibreren**

Bij het calibreren beweegt het meetinstrument met een vooraf ingestelde snelheid omhoog naar een eveneens vooraf ingesteld coördinaat. Elke 50ms leest hij de analoge ingang uit waar de versterker van de fotodiode op aangesloten is. Op deze manier kan bepaald worden bij welke hoogte hij de beste uitslag heeft. Wanneer er nu een product met hoogte X onder het meetinstrument wordt gelegd en deze zou de eerder beschreven actie herhalen, zou het bij dezelfde hoogte boven het voorwerp een beste uitslag geven. Omdat het voorwerp echter een hoogte X heeft zal het dus wel op een ander Z-coördinaat plaats vinden. Het verschil in beide Z-coördinaten is dan de hoogte van het product. Wanneer dit proces op verschillende X en Y coördinaten herhaald wordt kan er zo een hoogtemap ontstaan van het product. In de Bijlage B is de sourcecode te vinden van deze thread met commentaar inbegrepen.

### **XY-Scanner**

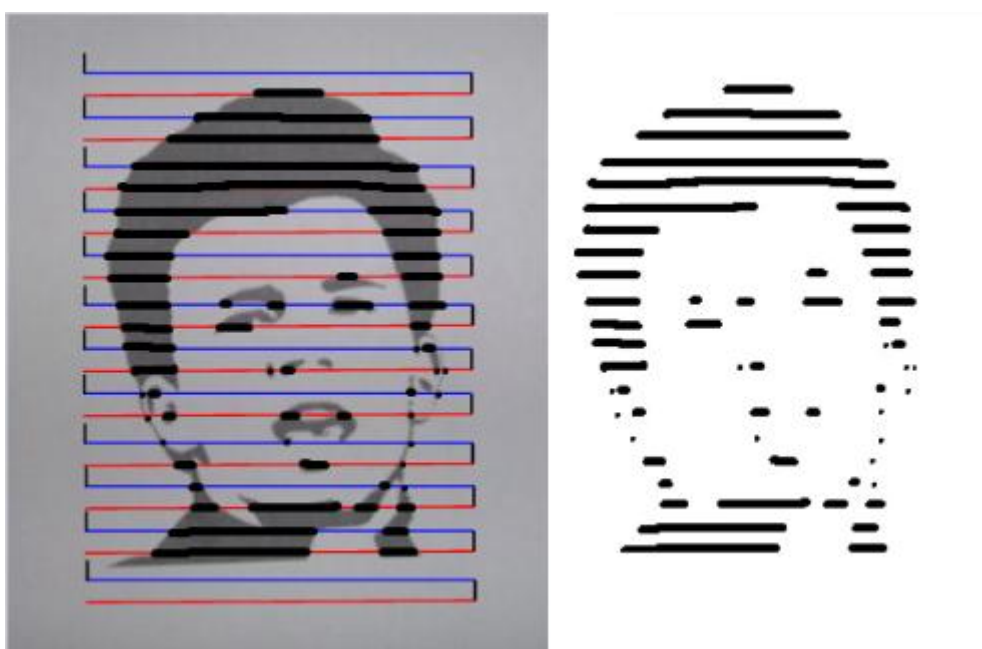
De XY-scanner scant een gebied af in de X en Y richting. Deze functie is met name gemaakt voor test doeleinden, wanneer er bijvoorbeeld een wit papier met een afbeelding erop wordt geplaatst in de CNC-machine, kan deze worden ingescand met behulp van de laser. De laser zal zich over het te scannen gebied bewegen en constant de intensiteit meten van het weerkaatste licht. Voordat de meting begint kan er worden aangegeven welke waarde de neutrale ondergrond heeft (in dit geval dus een stuk papier zonder inkt) en kan de scanner boven een gebied met inkt worden gezet om aan te geven hoeveel intensiteit er daar gemeten wordt. Het programma berekent dan met een formule vanaf welke gemeten waarde deze moet aanslaan. Wanneer de scanner zich boven inkt bevindt, schrijft deze zijn positie weg naar een bestand. Op deze manier ontstaat er een array aan coördinaten en als die dan geplot worden (in dit geval simpelweg met Microsoft Excel) ontstaat de ingescande afbeelding.

Figuur 13 en Figuur 14 tonen aan hoe dit scanproces werkt. Wanneer een afbeelding onder het meetinstrument wordt gelegd zal deze het traject afleggen welke te zien is in Figuur 13.



Figuur 13 Voorbeeld van een traject bij de XY-Scanner. De rode lijn is een beweging in de positieve x-richting en de blauwe lijn in de negatieve x-richting. De zwarte lijntjes zijn de stappen in de Y-richting.

Tijdens dit traject wordt er dus gereageerd op de inkt. Dit zorgt ervoor dat de laser wordt aangeslagen op gebieden met inkt en wordt het resultaat van Figuur 14 bereikt.



Figuur 14 Wanneer de coördinaten worden opgeslagen tijdens een beweging waar de inkt zich bevindt, zal er volgens deze voorspelling de illustratie ontstaan die rechts te zien is. Hierbij geldt: hoe kleine de stapjes in de y-richting hoe nauwkeuriger het beeld overeen komt.

Met de GUI kan worden ingegeven in welk gebied gemeten moet worden. Er kan een lengte, hoogte en snelheid worden ingevoerd. Verder is er een “Outside segment” en “Inside segment” knop te vinden. De bedoeling van deze twee knoppen is dat er eenmalig alvorens de scan begint, de laser boven een stuk papier wordt gezet waar geen inkt op zit. Op dat moment moet er op de knop

“outside segment” gedrukt worden. De waarde van de analoge ingang wordt op dat moment verstuurd naar de thread van de XY-Scanner. Hetzelfde moet gebeuren met “Inside segment” wanneer de laser zich boven inkt bevindt. The thread rekent dan zelf uit bij welke waarde van de analoge input hij moet registreren als inkt.

De sourcecode van de XY-Scanner is terug te vinden in Bijlage C.

## Hoofdstuk 5: Metingen

In dit hoofdstuk komen verschillende metingen aan de orde. Er zijn metingen verricht om het gedrag van het meetinstrument te onderzoeken en ook zijn er metingen verricht om te kijken hoe nauwkeurig iets kan worden ingemeten. Eerst zullen metingen aan de orde komen waarbij er puur wordt gekeken naar de stabiliteit van de laser. Vervolgens zijn er metingen verricht om te kijken met welke snelheid er gemeten kan worden en wat de invloeden daarvan zijn op de meetresultaten. Daarna zal het effect van ruis worden behandeld. De vertraging die in het meetsysteem zit komt ook aan bod en twee voorbeeld metingen gemaakt met een XY-Scan en een Calibratie meting zullen besproken worden en gekeken naar de nauwkeurigheid hiervan.

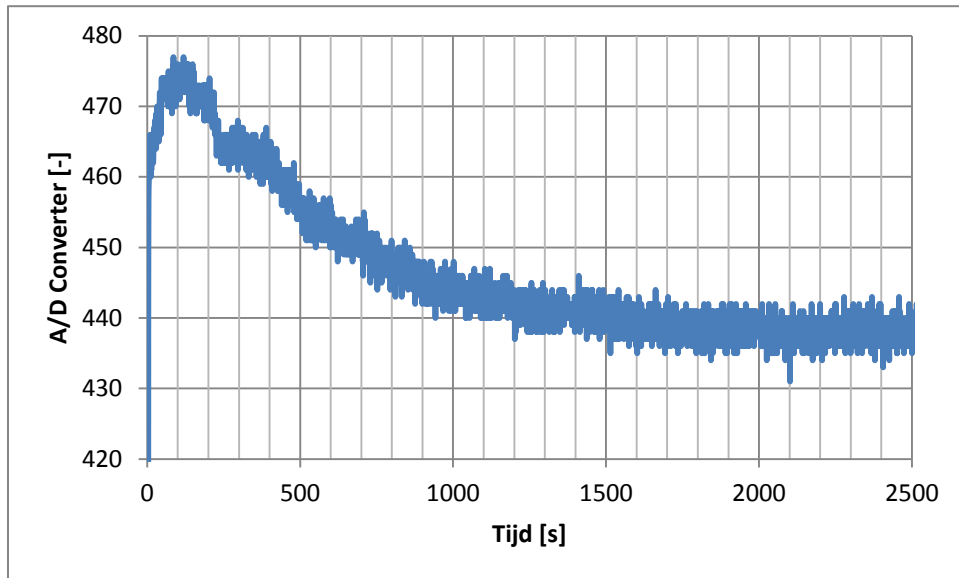
### *Stabiliteit van de laser*

Het gedrag van de laserdiode kan invloed hebben op de metingen. Uit de datasheets van de meeste laserdiodes staat vaak niet het gedrag beschreven als functie van de tijd. De laserdiode zal tijdens de werking warmte ontwikkelen en deze warmte zal invloed hebben op de laserwerking.

Het gedrag als functie van de tijd is bepaald met een experiment waarbij de laser en fotodiode op een vaste afstand worden gehouden van een voorwerp en de fotodiode constant wordt uitgelezen.

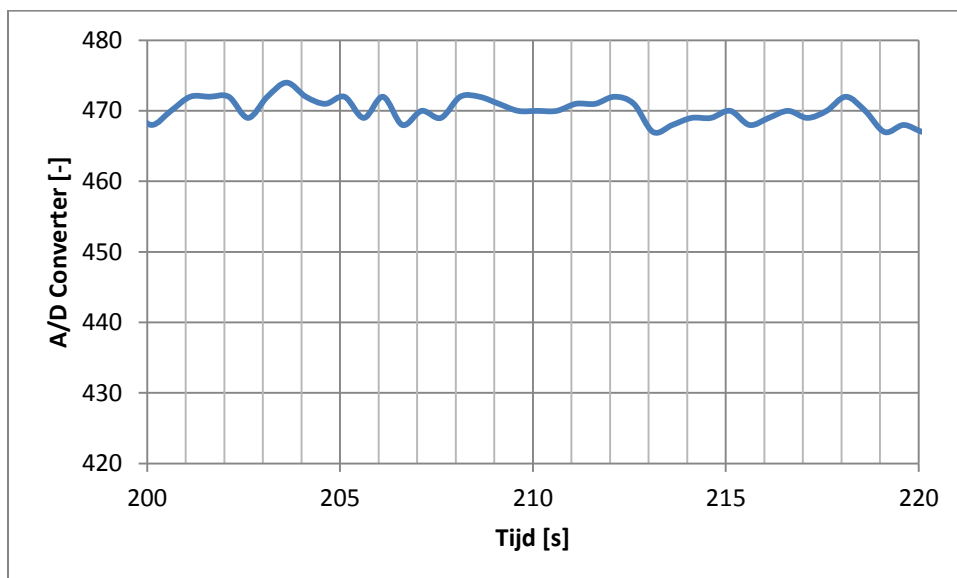
Door de laserdiode te activeren, is als functie van de tijd te zien wat er gebeurt met het uitgangsvermogen van de laserdiode.

Ter vergelijking is dit experiment uitgevoerd met twee verschillende laserdiodes. De ene laserdiode was de ADL-65075TR/L wat een 7 mW losse laserdiode is. De andere laserdiode was de PLP6501AR, een 1 mW laserdiode in een behuizing. Resultaten van de ADL-65075TR/L zijn te vinden in de Bijlage E. De laserdiodes zijn in allebei stabiel na enige tijd, maar als het vermogen hoger is veroorzaakt dat een groter verschil in het startvermogen en het stabiele vermogen. In de bijlage is te zien dat bij de ADL-65075TR/L minimaal 300 seconde gewacht moet worden om een signaal te hebben wat stabiel genoeg is om metingen mee te doen. Wat ook opviel bij deze losse laserdiode was dat een luchtstroming welke bijvoorbeeld ontstaat door de laserdiode te bewegen met een snelheid van ongeveer 1 cm/s al genoeg was om invloed te hebben op het uitgangsvermogen. Aangezien de laser tijdens de meting constant beweegt zou dit dus ook nadelige invloed kunnen hebben op de meting. Voor een geschikte laserkeuze moet het voorgaande beschreven dus in acht worden genomen. De PLP6501AR had geen merkbaar verschil op het uitgangsvermogen bij een beweging en ook was deze vrijwel meteen stabiel genoeg en klaar voor metingen. In Figuur 15 is het uitgangsvermogen te zien als functie van de tijd.



Figuur 15: Het gedrag van het uitgangsvermogen van de laser als deze wordt ingeschakeld. Het duurt 1500s voordat het signaal stabiel is,

Figuur 15 toont aan dat er ongeveer 200 seconde gewacht moet worden voordat er gemeten kan worden. Het signaal is pas na 1500 seconde echt stabiel maar na 200 seconde is het al stabiel genoeg om te meten. Dit toont Figuur 16 aan, waarbij te zien is dat het signaal constant (op de ruis na) is in een tijdsperiode van 20 seconde. Het gaat erom dat tijdens een meting het uitgangsvermogen niet fluctueert. Dus wanneer de meting binnen 20 seconde plaats vind (wat ruim is), ondervindt de meting minimale afwijkingen door het traag afnemende vermogen van de laser.



Figuur 16: Voor een klein tijdsbestek is het uitgangsvermogen van de laser constant te noemen.

## Ruis

In Figuur 15 en Figuur 16 was al te zien dat er ruis aanwezig is. Om een indruk te krijgen van hoe groot de ruis daadwerkelijk is wordt er van Figuur 15 gekeken bij het gebied van 1500 seconden tot 2500 seconden. Met de "LIJNSCH"-functie van Microsoft Excel is bepaald wat de standaard afwijking in de y-waarde is op dat gebied. De LIJNSCH-functie van Excel voert een regressie berekening uit met behulp van de kleinste kwadraten methode. Om een goede benadering te krijgen van de echte foutwaarde daarin is de standaard afwijking vermenigvuldigd met 3. Dit gaf een afwijking van 5 in de y-waarde. Dit betekent in feite dat de ruis op het stabiele signaal 5 bedraagt bij een waarde van 440.

De ruis ontstaat in de elektronische componenten en door omgevingsinvloeden. Bij omgevingsinvloeden is vooral het omgevingslicht een storende factor. De metingen zijn verricht in een ruimte waar ook daglicht is, dit is met opzet zo gedaan omdat in de praktijk dit ook vaak het geval is. Daglicht is geen constante lichtbron wat dus ruis zal veroorzaken. Maar ook het licht/vermogen van de laser zelf is niet constant. De laserwerking verloopt volgens een stochastisch proces. De situatie waarin de meetopstelling zich bevindt is ook verreweg van ideaal. Op een cnc-besturing zitten stappenmotoren die elektrische velden veroorzaken. Bij het meten van kleine signalen zal dit dus ook ruis opleveren in de kabels en elektrische componenten van de meetopstelling.

Bij de ruis die ontstaat in de elektrische componenten dragen vooral de weerstand  $R_f$  en condensator  $C_f$  bij, welke beide gebruikt worden voor de transimpedantie versterker. (hierover meer bij "snelheid systeem"). Er zal ook thermische ruis ontstaan in weerstand  $R_f$ . De thermische ruis die ontstaat bij een weerstand is evenredig met de grootte van de weerstand.

### Adviezen om de ruis te minimaliseren bij de opstelling:

- Probeer metingen te verrichten in een zo donker mogelijke ruimte.
- Gebruik afgeschermd kabels om de data over te brengen.
- Plaats de elektrische componenten zo ver mogelijk uit de buurt van de stappenmotoren.
- Probeer de weerstand  $R_f$  zo klein mogelijk te houden
- Gebruik elektrische componenten die zelf zo min mogelijk ruis veroorzaken, dit is vooral bedoeld bij de keuze van de OP-AMP, hierin is veel keus en verschillen de ruiswaardes enorm per type.

## Snelheid van het systeem

De snelheid waarmee het systeem reageert op een verandering van de intensiteit is belangrijk voor het afstellen van de juiste snelheid waarmee gemeten kan worden. Er zijn meerdere factoren van invloed op de snelheid waarmee gemeten kan worden. Logische aspecten zijn de reactietijden van de onderliggende elektrische componenten, in dit geval zijn dat de foto-diode in combinatie met de transimpedantie versterker en de AD-converter op de microcontroller. Buiten deze componenten moet het software pakket (en dus ook de computer hardware) ook de snelheid aan kunnen.

De datasheets van de elektrische componenten tonen enkele reactietijden, maar deze zijn afhankelijk van de situaties dus er kan niet voor dit specifiek geval uit de datasheet worden gehaald



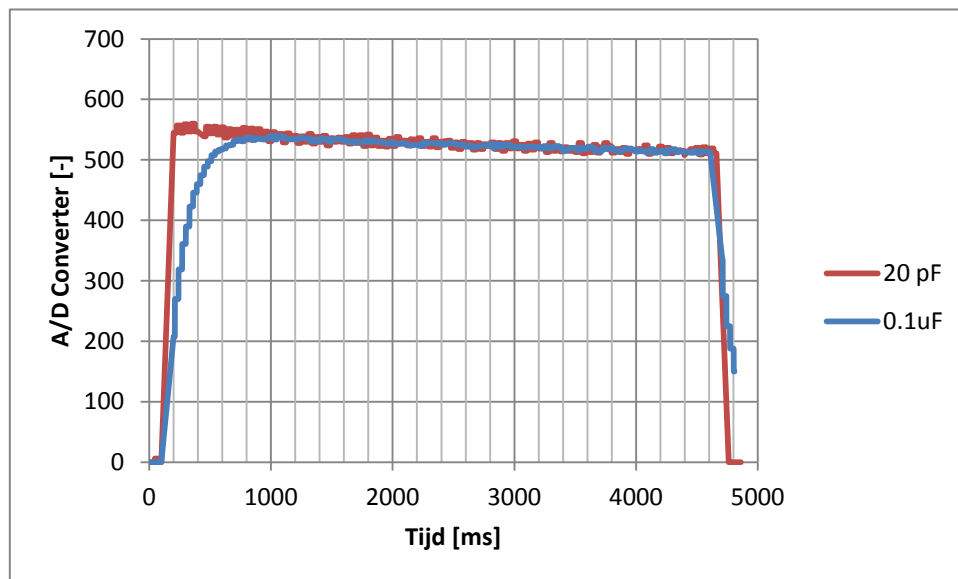
hoe snel de elektrische componenten reageren. Wel kan er een schatting worden gemaakt van in welke orde van grootte gedacht moet worden. Informatie uit de datasheets is te zien in Bijlage F.

Hieruit volgt dat de snelheid waarmee het systeem reageert in de enkele microseconden ligt. Uit experimenten met de software is echter gebleken dat er maar elke 50 ms gemeten kan worden. Wanneer de software werd ingesteld om elke ms te meten gebeurde dit wel (met af en toe een vertraging) maar werd er steeds pas een nieuwe waarde gemeten na ongeveer 30ms. Omdat het niet altijd precies 30ms was en er af en toe een kleine vertraging in zat van enkele ms is er vanuit gegaan dat het systeem elke 50ms een betrouwbare meting kan verrichten. Een eenvoudig experiment wees uit dat dit niet lag aan de meetopstelling of de software maar aan de USB CNC CPU.

Een variabel in te delen aspect bij de meetopstelling is de combinatie van de weerstand  $R_f$  en  $C_f$ . De weerstand is bepalend voor de mate van versterking en de condensator gaat uitschieten van het signaal als mede het oscilleren tegen. De mate van versterking moet zo optimaal mogelijk zijn, hierdoor wordt de condensator dus afgestemd wanneer er een juiste waarde voor de weerstand is gekozen. De condensator en weerstand hebben samen een  $\tau$ -tijd. Deze moet zo klein mogelijk gehouden worden zodat het systeem zo snel mogelijk kan reageren. Omdat er al is bepaald dat er elke 50ms gemeten wordt, zou de  $\tau$ -tijd dus logischerwijs kleiner moeten zijn dan 50 ms.

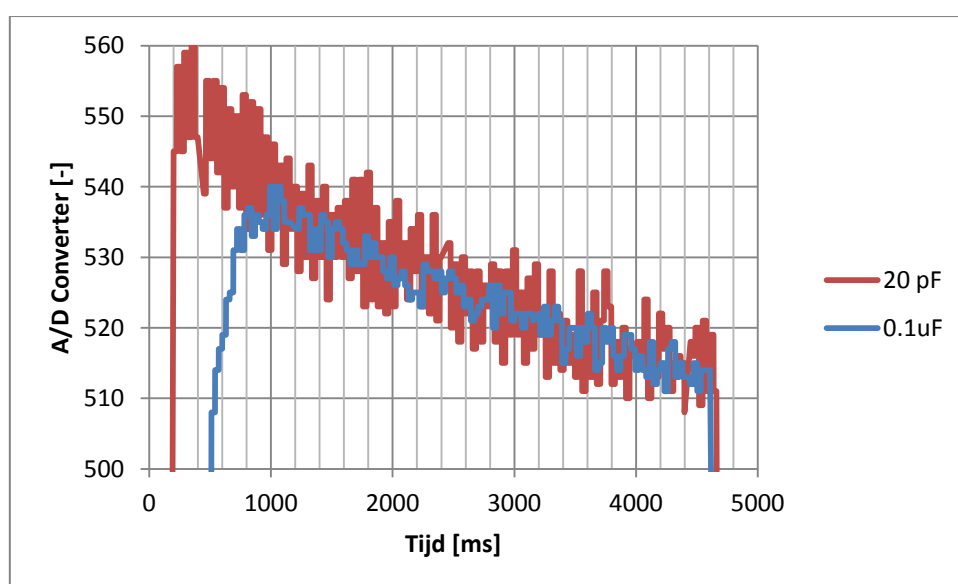
Een te kleine  $\tau$ -tijd brengt echter weer veel ruis met zich mee. Er moet dus een compromis gezocht worden waarbij zo min mogelijk ruis ontstaat en het systeem toch met een gewenste snelheid reageert. Hiervoor is een testopstelling gemaakt waarbij verschillende condensatoren zijn doorgemeten. De testopstelling bestond uit de meetopstelling maar dan was de laserdiode vervangen voor een normale LED en direct op de fotodiode gericht. Er is een programma geschreven die elke milliseconde een meting verricht en de LED aanzet na een bepaalde tijdsduur en weer uitschakelt. Hierdoor is te meten hoe snel het systeem reageert op een plotseling vermogen van de LED.

Er zijn vele condensatoren doorgemeten en om aan te geven wat het verschil was is in Figuur 17 een 20 pF condensator vergeleken met een 0,1  $\mu$ F. Beide condensatoren zijn gemeten onder dezelfde omstandigheden.



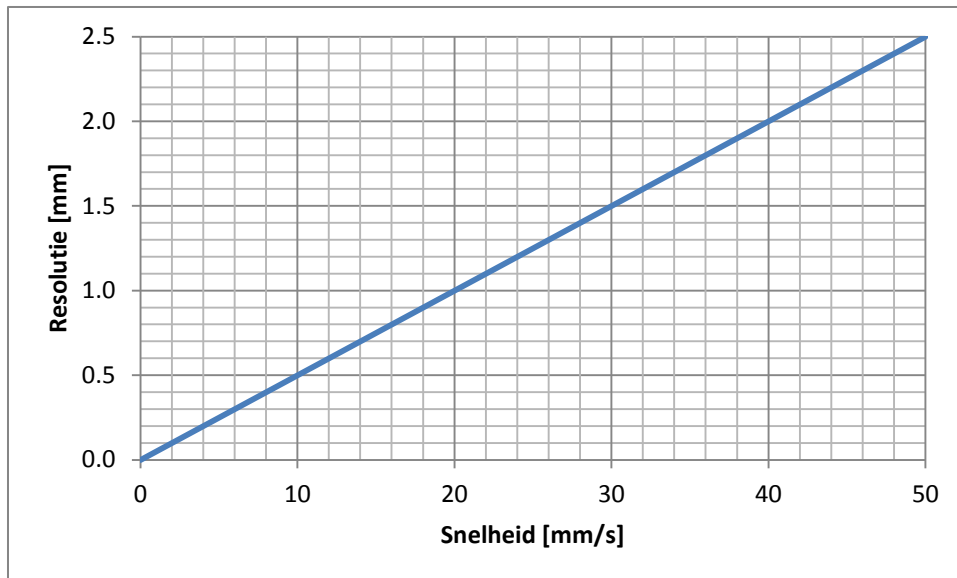
Figuur 17 Meting verricht met twee verschillende condensatoren. Hierin is te zien dat de kleine condensator sneller reageert, maar ook een grotere overshoot heeft, aan deze meting is een guide-for-the-eye toegevoegd.

De  $\tau$ -tijd voor de 20 pF en de 0,1  $\mu$ F zijn berekend met formule [5] en zijn respectievelijk 0,10 ms en 500 ms. Deze waarden zijn in overeenstemming met wat te zien is bij de meting. Er werd gewenst een systeem dat zo snel mogelijk reageert maar toch gaat de voorkeur uit naar de condensator van 0,1  $\mu$ F. De ruis en overshoot is minder aanwezig dan bij de 20 pF condensator, dit is te zien in Figuur 18. De  $\tau$ -tijd van 500 ms lijkt echter niet te voldoen, maar er moet gerealiseerd worden dat de meting in kleine stapjes verricht wordt waarbij de intensiteit oploopt en niet instantaan verhoogd wordt. Hierdoor voldoet een  $\tau$ -tijd van 500 ms dus toch.



Figuur 18 Zelfde grafiek als figuur 1 maar met aangepaste y-as waardoor duidelijk te zien is dat de ruis kleiner is bij een grotere condensator, aan deze meting is een guide-for-the-eye toegevoegd.

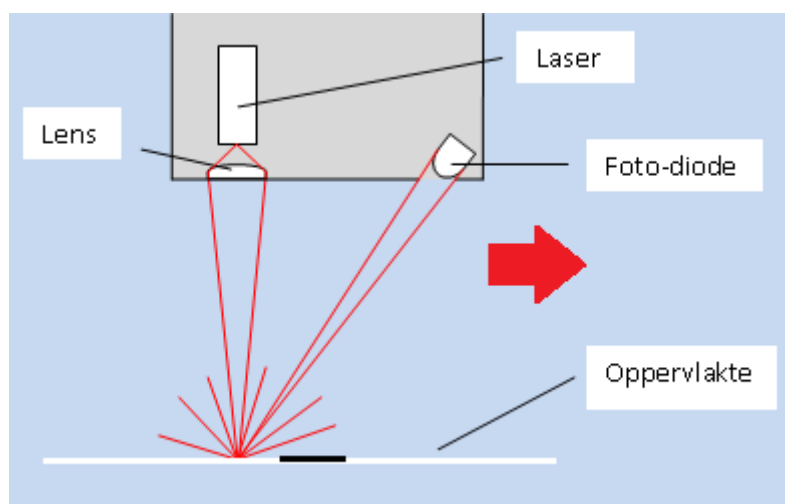
De snelheid van het systeem heeft directe gevolgen voor hoe snel er gemeten kan worden. Er wordt een sample tijd van 50 ms gehanteerd wat betekent dat er met 20 Hz samples worden genomen. Als de snelheid waarmee bewogen wordt bekend is kan worden berekend welke resolutie het systeem heeft. Hieruit volgt een lineair verband.



**Figuur 19** Deze grafiek toont aan dat er een lineair verband is tussen de snelheid waarmee wordt gemeten en welke resolutie er gehaald kan worden.

Om de software en het meetinstrument te testen is er een programma geschreven waarmee een XY-Scan gemaakt kan worden.

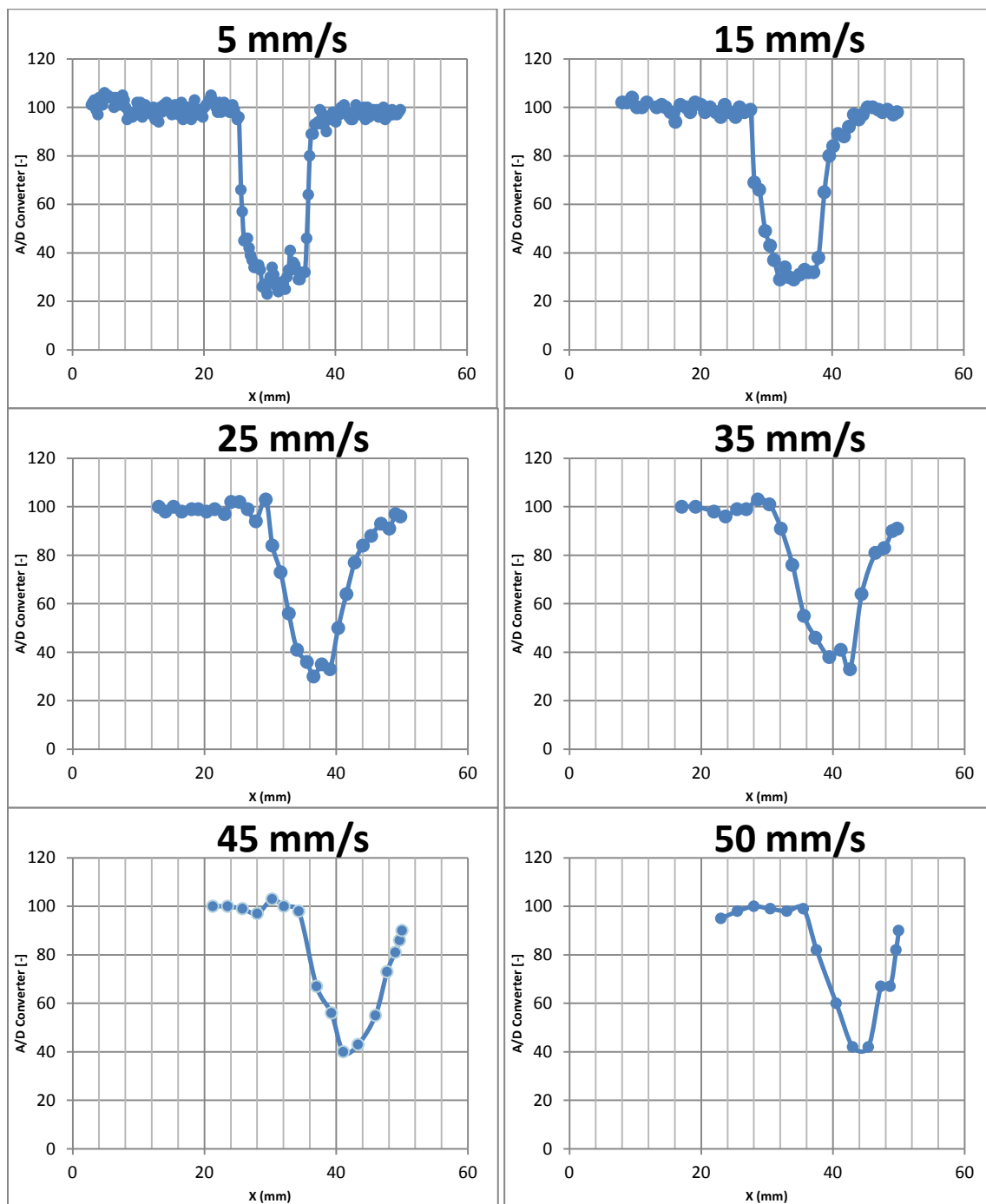
Hierbij houdt de meetopstelling een vaste hoogte aan en beweegt alleen in de X en Y richting over een voorwerp. In dit geval is het een wit A4 papier met daarop een gedrukte afbeelding. Dit proces is te zien in Figuur 20.



**Figuur 20** Door de meetopstelling op een vaste hoogte te houden kunnen veranderingen in het signaal duiden op absorptie.

Voor de metingen is een dikke zwarte geprinte streep gebruikt van 10mm. De laser is met verschillende snelheden over de streep heen bewogen. Steeds over hetzelfde stuk van de streep. In

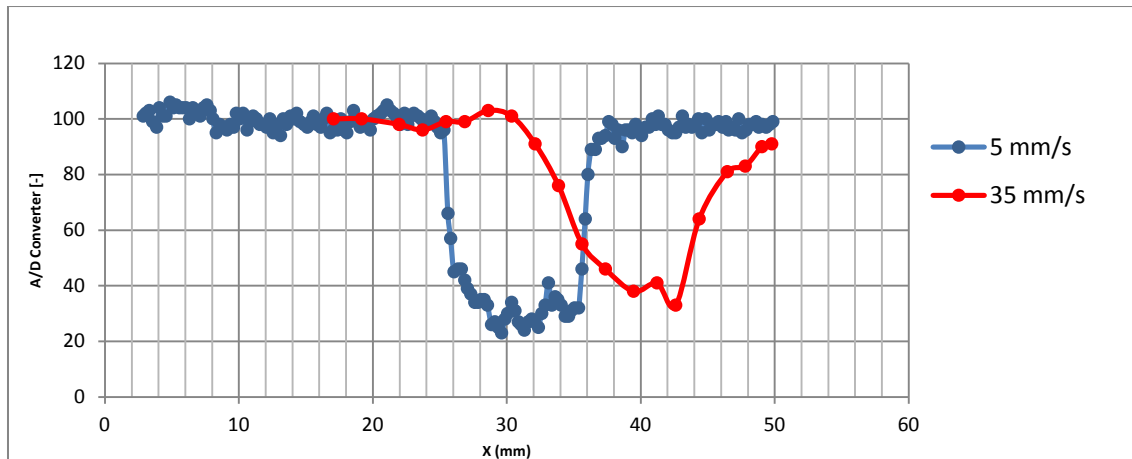
plaats van alleen waarden weg te schrijven bij de inkt, werd in dit geval de waarde elke 50 ms weggeschreven naar een bestand om het gedrag van de meetopstelling te analyseren. Het resultaat van deze metingen is te zien in Figuur 21.



**Figuur 21** Hier zijn 6 verschillende grafieken te zien die steeds hetzelfde meettraject hebben doorlopen maar met verschillende snelheden.

Uit de figuren kunnen twee belangrijke conclusies worden getrokken. Bij een lage snelheid is te zien dat het verloop naar de inktvlek snel verloopt. Het signaal heeft maar kort nodig om weg te zakken waardoor er duidelijker bepaald kan worden waar de inktvlek begint en waar deze eindigt, met andere woorden houdt dit in dat de nauwkeurigheid dus hoger is bij lagere snelheden. Dit is ook

logisch aangezien er steeds elke 50 ms wordt bewogen. De afstand die bewogen wordt in de tijd van 50 ms is al direct de minimale nauwkeurigheid die nog gehaald kan worden. Er is ook duidelijk te zien dat er sprake is van vertraging in het signaal. Hoe sneller er bewogen wordt hoe verder het dal zich verplaatst in de positieve x richting. Ter vergelijking, bij een snelheid van 15 mm/s begint het dal bij ongeveer 24mm. Terwijl bij een snelheid van 50mm/s het dal pas begint bij 38mm. Met deze vertraging moet uiteraard rekening worden gehouden. Als de vertraging altijd hetzelfde is zou hier makkelijk voor gecorrigeerd kunnen worden.



**Figuur 22** De twee meetlijnen van 5 mm/s en 35 mm/s zijn hier bij elkaar gevoegd waarbij goed het verschil te zien is in nauwkeurigheid en de verschuiving die plaats vindt ten gevolge van de snelheid.

De meetgegevens van dit experiment bevestigen het lineaire verband van Figuur 19. Om dus te kunnen meten met een nauwkeurigheid van 0,1 mm mag de machine niet sneller bewegen dan 2 mm/s.

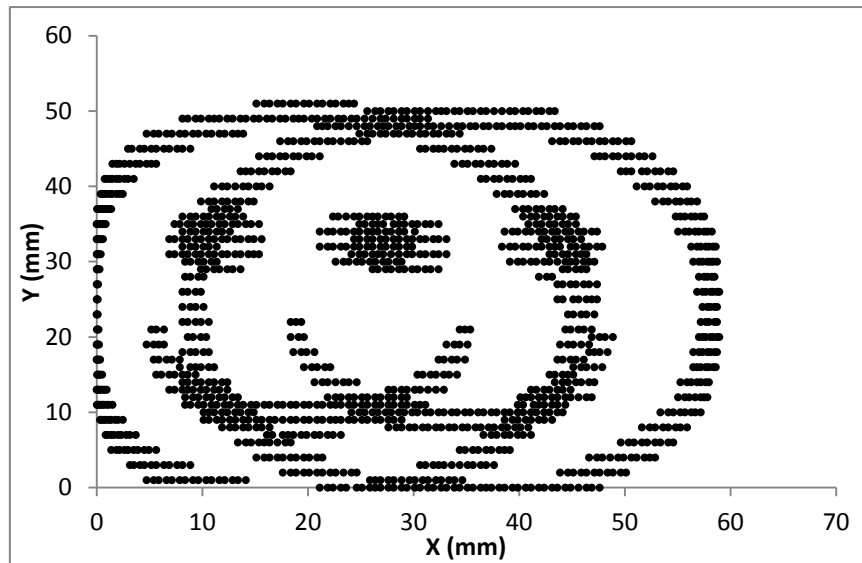
### Vertraging

Om de vertraging te visualiseren is de XY-Scan uitgevoerd zoals eerder in het verslag beschreven. Er is een eenvoudige afbeelding uitgeprint en deze is ingescand volgens de procedure. De meting is verricht bij een snelheid van 25 mm/s en de stapgrootte van y was 1 mm.



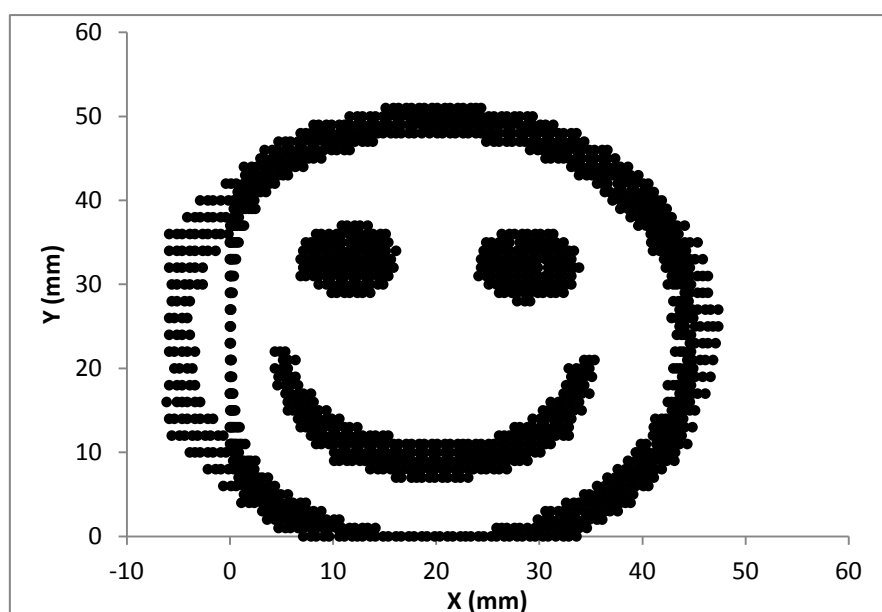
**Figuur 23** De XY-Scanner in actie wanneer deze het plaatje inmeet wat gebruikt is voor de beschreven meting.

De scan leverde het volgende resultaat op:



Figuur 24 Het resultaat van de meting, het figuur wordt in twee verschillende richtingen vertraagd waardoor het dubbel wordt afgebeeld.

De beweging in positieve x-richting krijgt nu een vertraging en de bewegingen in de negatieve x-richting ook. Dit zorgt ervoor dat alle punten in beide richtingen dus vertraagd worden en levert een afwijkend resultaat op. Als de vertraging constant is zou uit de meetgegevens van Figuur 24 met een kleine aanpassing toch een goed resultaat te halen moeten zijn. Om dit aan te tonen heeft er een correctie plaatsgevonden op alle meetpunten die gemaakt zijn tijdens de positieve x-as beweging. Bij elke positie is steeds dezelfde waarde afgetrokken, in dit geval 14mm. Dit leverde het plaatje van Figuur 25 op.



Figuur 25 De meetgegevens van het experiment maar ditmaal gecorrigeerd voor de vertraging van het systeem.

Dit geeft visueel aan dat alles goed samenvalt bij een constante correctie factor. Bij de randen zit enige onregelmatigheid maar dit heeft niets te maken met de vertraging, dat lag aan het meetbereik bij het experiment.

Omdat er hier voor 14mm gecorrigeerd werd betekent dat het systeem dus in 1 richting 7 mm later reageert. Met een snelheid van 25 mm/s komt dit er dus neer dat de vertraging van het systeem circa 0,28 s. Visueel gezien is deze vertraging geen probleem, deze is constant genoeg om er eenvoudigweg voor te corrigeren. Bij een perfect constante vertraging zou het ook voor een hoogtemeting geen problemen opleveren. Bij het calibreren zal er weliswaar een te hoge calibratiehoogte uit komen, maar wanneer de hoogte van een product wordt gemeten komt er ook een waarde uit met precies dezelfde afwijking als bij de calibratiehoogte (op voorwaarde dat het systeem met dezelfde snelheid beweegt). Onnauwkeurigheden kunnen wel ontstaan doordat de vertraging niet volledig constant is maar kleine fluctuaties heeft of door de snelheid van de CNC-besturing die kleine afwijkingen heeft. De nauwkeurigheid van de CNC-besturing hangt volledig af van de machine zelf en de besturing dus daar is dus weinig over te zeggen, er wordt wel vanuit gegaan dat deze dusdanig nauwkeurig is dat dit in vergelijking met andere storende factoren te verwaarlozen is.

Om onnauwkeurigheden in meetwaarden tegen te gaan zou theoretisch gezien de snelheid waarmee gemeten wordt verlaagd kunnen worden. Als rekenvoorbeeld, de vertraging van het systeem schommelt tussen de 0,25 s en de 0,35 s. Dat is een verschil van 0,10 s. Wanneer een besturing met een snelheid van 25 mm/s zou bewegen, wordt de onnauwkeurigheid dus al 2,5 mm. Wanneer de machine met 1 mm/s zou bewegen is dit slechts 0,1 mm.

### *Meting met de XY-Scanner*

Om te zien waartoe de XY-Scanner in staat is, is er een complexere afbeelding uitgeprint om die in te scannen. Vanwege de vertraging in het systeem is de software dusdanig aangepast dat de scanner alleen nog maar in de positieve x-richting metingen verricht. Scanmethode is nog steeds hetzelfde als die van Figuur 13 alleen beweegt hij nu met de ingestelde snelheid op de positieve x-as richting en gaat hij steeds met de maximale snelheid op de negatieve x-as beweging. Om geen meetpunten te verliezen is de stapgrootte van y logischerwijs wel gehalveerd.

Het resultaat van de scan is te zien in Figuur 26.



**Figuur 26** Links is het originele plaatje wat is uitgeprint en vervolgens gescand. Het resultaat is het plaatje rechts.

De scan is uitgevoerd met een snelheid van 2,0 mm/s. Dit zou de maximale snelheid zijn om met de huidige configuratie een nauwkeurigheid van 0,1 mm te halen. Het is echter alleen visueel te beoordelen of de scan gelukt is. In dit geval is te constateren dat de details en de verhoudingen overtuigend zijn. Maar wanneer er bijvoorbeeld wordt gekeken naar de rechter kaaklijn van het ingescande hoofd blijkt dat deze wegvalt bij de scan. Deze kaaklijn was op de afbeelding dikker dan 0,1 maar is toch niet bij de scan verschenen. Hierbij spelen 2 factoren een rol. De gevoeligheid waarmee hij is afgesteld en de grootte van de gefocusseerde laserbundel. De gevoeligheid wordt nu bepaald door een empirisch samengestelde formule. Deze kan natuurlijk aangepast worden om de meting gevoeliger te maken, maar dat heeft ook weer meetpunten tot gevolg die ongewenst zijn. De laserpunt is bij de huidige situatie ook nog relatief groot. Op het oog gezien is deze naar schatting 0,5 mm. Wanneer een grote bundel een dun lijntje wil detecteren zal deze dus gevoeliger moeten worden ingesteld.

### *Calibratie*

Voor een hoogtemeting is de calibratie een essentieel onderdeel. De bedoeling bij een calibratiemeting is dat de meetopstelling zo dicht mogelijk bij een oppervlakte begint te meten en alsmaar omhoog beweegt. De positie waarbij de hoogste waarde wordt gemeten is dan de calibratiehoogte. Om te zien hoe dit in de praktijk werkt zijn er veel metingen verricht om het gedrag van de meetopstelling te bepalen. Ook is er een theoretisch model gemaakt in C++ om te voorspellen wat er gemeten zou moeten worden. Dit model is terug te vinden in de Bijlage E.

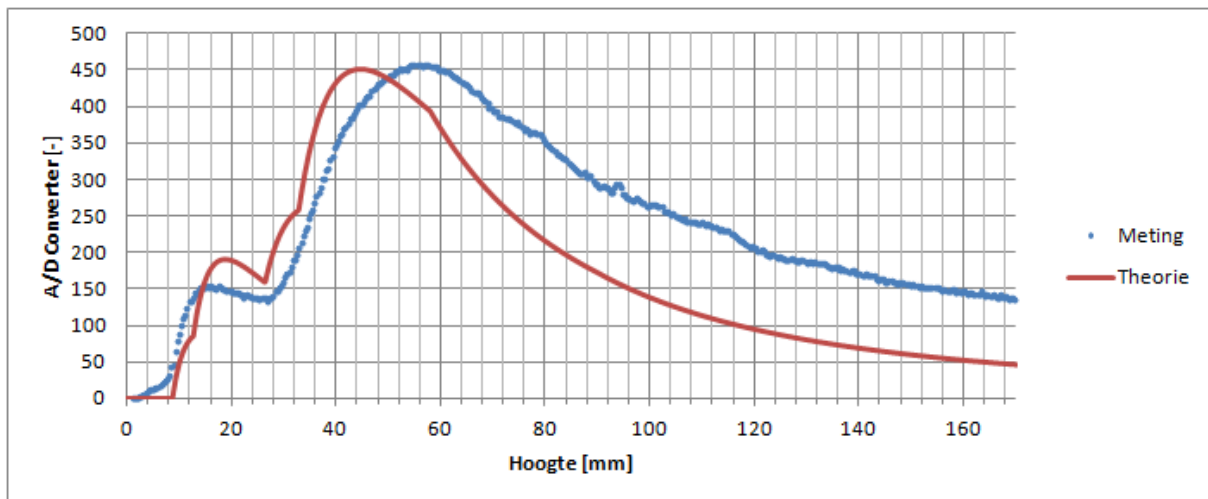
Het theoretische model is uitgevoerd met de volgende variabelen:

Vermogen laser	0.6mW*
Hoek diode	10 graden
Weerstandswaarde op-amp	5000000 Ohm
Afstand laser en fotodiode	1.5 cm

\*Merk op dat de 0,99 mW laser dus niet op vol vermogen was ingesteld tijdens deze meting.

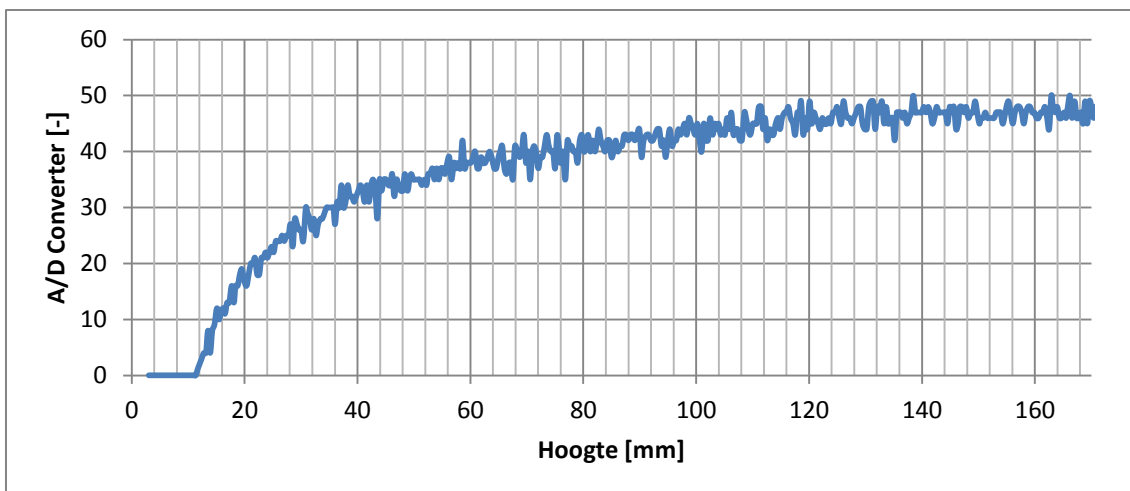
De meting werd verricht vanaf een wit A4 papier op een hoogte van 1mm t/m 170mm. De meetwaarden en de theoretische waarden zijn te zien in Figuur 27.





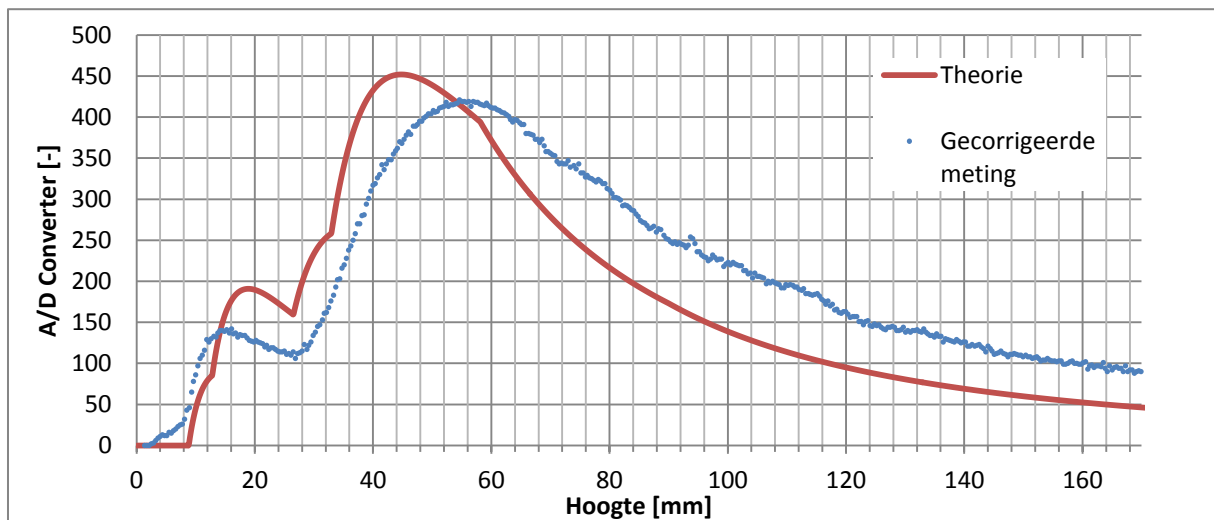
Figuur 27 De grafiek toont zowel de gemeten waarde als de theoretisch voorspelde waarden.

Voordat Figuur 27 geanalyseerd kan worden moeten er eerst nog enkele correcties op worden uitgevoerd. Allereerst was er achtergrondstraling aanwezig tijdens de meting. Om hiervoor te corrigeren is exact dezelfde meting uitgevoerd onder dezelfde omstandigheden maar dan waarbij de laserdioden uit stond. Deze meting is te zien in Figuur 28. (Voor een hoogtemeting zou er normaal niet gecorrigeerd hoeven worden voor de achtergrondstraling, maar om de meetwaarden te vergelijken met de theoretische voorspelling is dit wel van belang).



Figuur 28 De achtergrond straling onder dezelfde omstandigheden als de meting uit Figuur 17. In het begin neemt de achtergrond straling toe maar gaat vervolgens naar een stabiele situatie.

Deze achtergrondstraling is simpel weg afgetrokken van het meetsignaal. Er is ook een correctie gemaakt voor de vertraging. Empirisch is bepaald dat de vertraging ongeveer 0,28 s bedraagt. Door deze te vermenigvuldigen met de snelheid van het systeem komt er een correctie voor de hoogte uit van 3,5 mm. Ook deze waarde is afgetrokken van de meetwaardes. Beide correcties leveren de volgende nieuwe grafiek op.

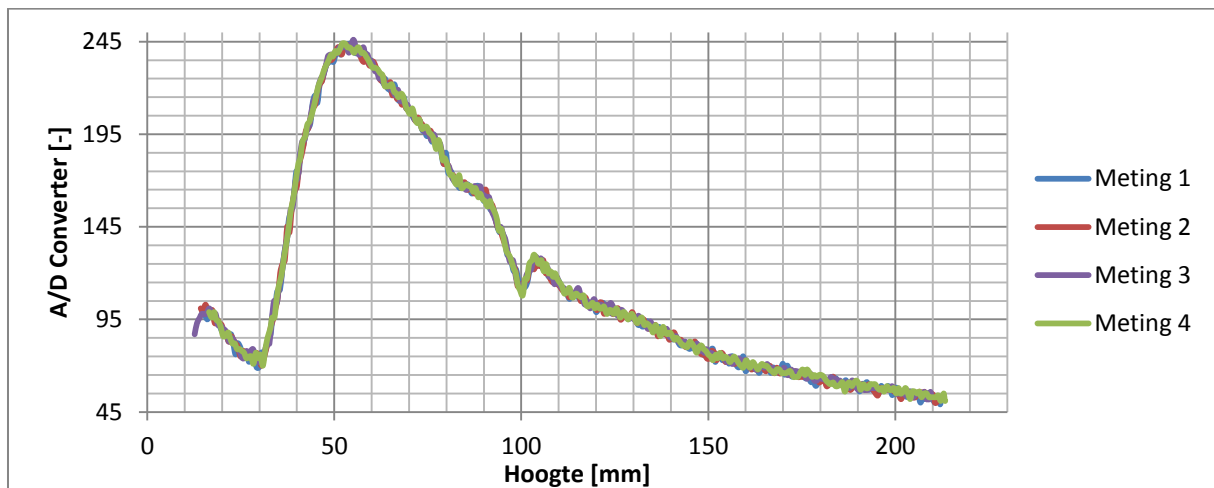


Figuur 29 Dezelfde meetgegevens als Figuur 27 maar dan met correcties uitgevoerd op de meetwaarden.

Ondanks beide correcties vertonen de meetwaarden nog steeds opvallend afwijkend gedrag. Te beginnen met de verschuiving. De correctie van de vertraging heeft de verschuiving wel gereduceerd maar er zit alsnog een afwijking in met de theorie. Vanaf circa 60 mm valt op dat de meetwaarden hoger liggen dan de theoretisch voorspelde waarden. Dit is ook opvallend, in het model is namelijk niet uitgegaan van verliezen. Het licht van de laser zal deels worden geabsorbeerd door het materiaal waar het opvalt en in geringe mate door de lucht. Ook zullen er verliezen optreden omdat het vel papier geen perfecte lambertse-straler is. Naar verwachting zouden dus de meetwaarden ten alle tijde onder de theoretische waarde moeten zijn. Waarom dit dus niet gebeurt is niet precies duidelijk. Het vermoeden is dat het aan de afstelling ligt van de laserdiode en de fotodiode. Om de theoretische waarden te laten kloppen zouden deze perfect uitgelijnd moeten staan. Dat wil zeggen dat de hoek die de fotodiode maakt evenwijdig is naar het middelpunt van de stip van de laser. Wanneer dit niet het geval is komt het licht onder een andere hoek de laserdiode binnen en kloppen de relaties en verhoudingen van het theoretische model ook minder nauwkeurig. Dit zorgt er dus voor dat het moeilijker te voorspellen is wat er wordt gemeten, desalniettemin kunnen er alsnog metingen verricht worden omdat deze afwijking natuurlijk constant is. De fotodiode is gevoeliger voor hoeken van inval rond de  $0^\circ$  graden. Dus wanneer er door slechte uitlijning deze hoek nooit gehaald wordt betekent dit wel dat de meting dus minder nauwkeurig kan worden uitgevoerd. Om deze reden is het streven naar een goede uitlijning dus toch van groot belang.

Een wat kleiner detail wat opvalt, is dat het theoretische model aantoont dat er pas vanaf 10 mm een signaal gemeten zal worden maar er wordt al sneller wat gemeten. Dit kan twee oorzaken hebben, achtergrondlicht zou eerder bij de fotodiode binnen kunnen komen dan het licht van de laser, dit is echter niet het geval want in Figuur 28 is te zien dat er pas achtergrondlicht wordt gemeten vanaf 12mm. De echte reden is al genoemd bij het hoofdstuk theorie. Voor dit model is gebruikt gemaakt van een benadering, de fotodiode wordt gezien als een oppervlakte wat mee kromt met de halve bol waarop het zich bevindt. In werkelijkheid heeft de fotodiode een gekromd oppervlakte tegengesteld aan die van de kromming van de halve bol. Op zeer kleine afstanden is de benadering dus niet nauwkeurig omdat de fotodiode zelf ook nog een lengte heeft. Dit is de reden dat er toch al eerder een signaal gemeten wordt.

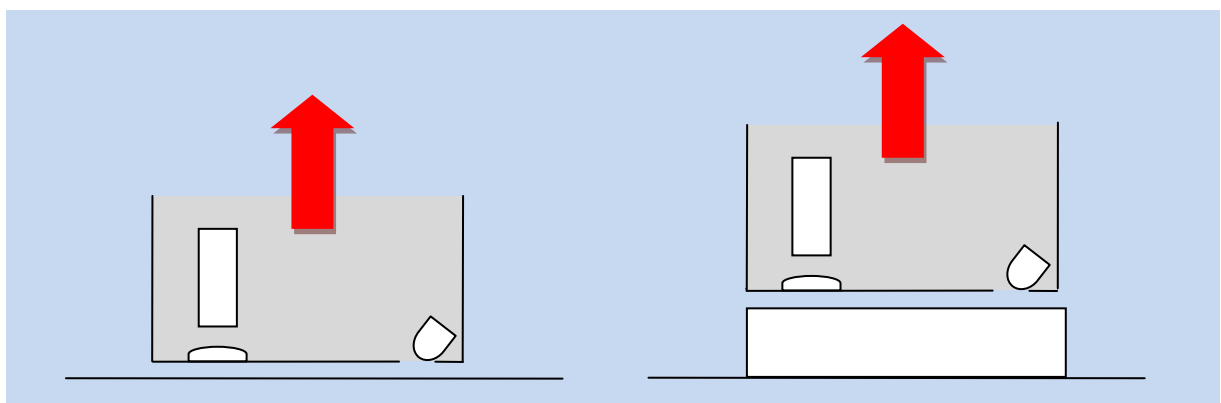
In de calibratielijnen van de voorgaande figuren was er geen sprake van, maar vaak werd er ook een ander verschijnsel waargenomen wat te zien is in Figuur 30. De lijn loopt minder mooi af na de top, bij 100 mm zit een duidelijke afwijking. De vraag is wat dit veroorzaakt en of het consequent is. Er kan worden geconstateerd dat het niet een toevalligheid is omdat dezelfde meting 4x is herhaald en iedere keer hetzelfde patroon zich voordoet. De reden van deze afwijking zit hem weer in de uitlijning van de laser. Als de laser niet perfect is uitgelijnd en dus niet loodrecht op het oppervlak eronder staat, zal de laserpunt veranderen van locatie wanneer de laser zich verder van het oppervlak afbeweegt. Als er dan, zoals in dit geval een velletje papier, daar toevallig net een kleine afwijking in zit heeft dat dus invloed op de curve. Er is hiermee wel aangetoond dat meting reproduceerbaar is.



Figuur 30 De grafiek toont 4x een herhaalde meetreeks. Hierbij valt op dat pieken die in de curve plaatsvinden bij elke meetreeks voorkwamen en dus niet toevallig waren.

### Inmeten van hoogte

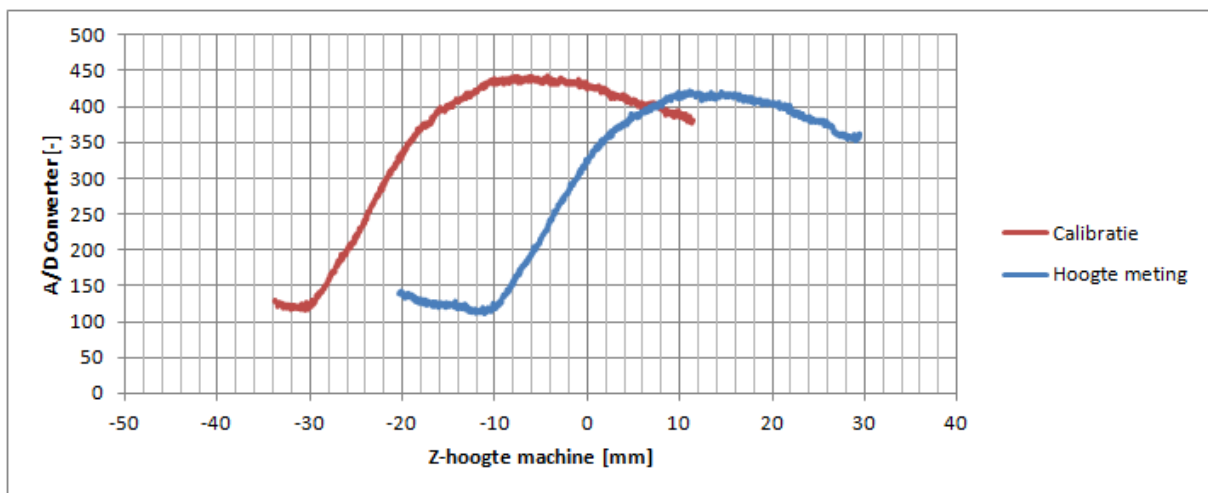
Voor het inmeten van een hoogte moet er eigenlijk twee keer een calibratie meting gedaan worden. Eén keer op de ondergrond en één keer wanneer de laser zich boven het in te meten product bevindt. Om materiaal eigenschappen te blokkeren is ervoor gekozen om een object van 20 mm hoogte in te meten met daarop een dun vel papier van 0,1 mm. Dus een totale hoogte van 20,1 mm. In het ideale geval wordt er gewoon gekeken bij welke positie de hoogste waarde wordt gemeten en daaruit kan dan de hoogte worden gehaald.



Figuur 31 Schematische weergave van het experiment. Links is de calibratie meting te zien waarbij de meetopstelling vanaf de tafel omhoog beweegt en de meting verricht. Rechts is er een voorwerp op de tafel geplaatst, de meetopstelling gaat nu meten vanaf vlak boven het voorwerp.

Bij de opstelling is er dus voor gekozen een vel papier op het in te meten voorwerp te leggen om er een lambertse-straler van te maken. Wanneer dit niet gebeurt treden er, afhankelijk van materiaalsoort, effecten op die veel vermogen absorberen of teveel in één richting reflecteren. Dit probleem mag niet genegeerd worden wanneer er daadwerkelijke scans mee uitgevoerd gaan worden, om tests uit te voeren is ervoor gekozen om dus alleen te werken met lamberste-stralers.

Beide metingen van zowel de calibratie als de hoogte meting zijn geplot wat te zien is in Figuur 32. Op de x-as staan werkstukcoördinaten, dit betekent dat in dit geval het nulpunt niet lag bij de eerste calibratiemeting. Dit is geen probleem omdat het hier puur gaat om het verschil van de calibratie en de hoogte meting.



Figuur 32 Hier zijn twee meetreeksen te zien van een calibratie meting. De hoogtemeting is gemeten bij een voorwerp met een hoogte van 20,1 mm.

Wanneer van beide meetreeksen de A/D Converter waardes gesorteerd worden van hoog naar laag levert dat het resultaat op van tabel 1 en tabel 2. Voor zowel de normale calibratie als voor de hoogte meting zijn in beide gevallen hieronder in een tabel aangegeven hoeveel de waarden van elkaar verschillen in dit geval. Er is maar een klein gedeelte van de totale tabel weergegeven om een impressie te geven.

Tabel 1: Een kleine selectie van de topwaardes van de calibratie meting.

Z-positie [mm]	A/D Converter [-]
-4.3375	443
-7.8625	442
-7.5625	442
-6.2875	442
-6.055	442
-8.2375	441
-6.5125	441
-6.4	441
-6.175	441
-5.905	441

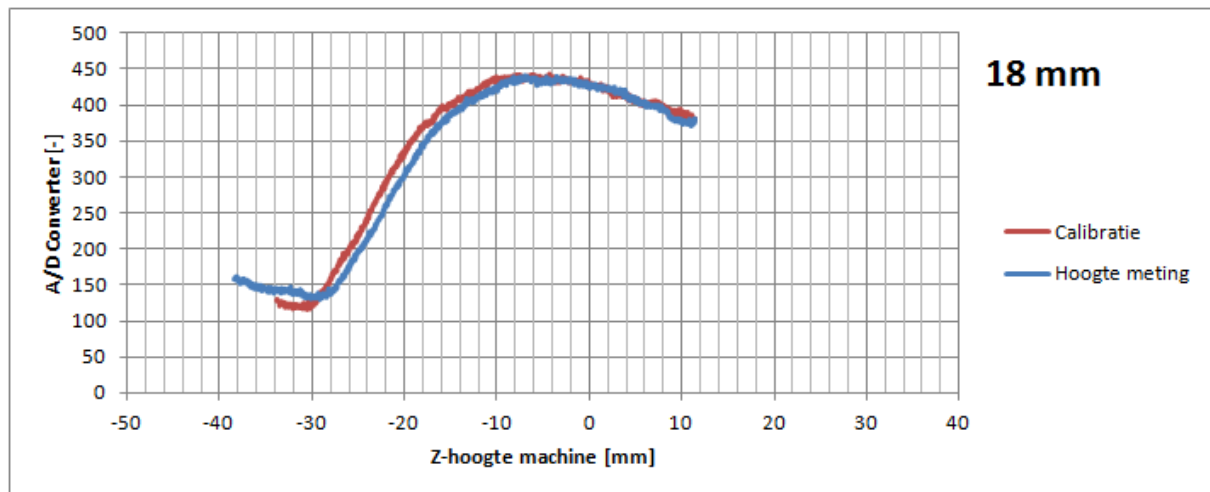
**Tabel 2: Een kleine selectie van de topwaardes van de hoogte meting**

Z-positie [mm]	A/D Converter [-]
11.065	423
11.1025	422
11.29	422
14.515	422
9.865	421
10.615	421
11.365	421
13.39	421
10.315	420
11.3275	420
11.8525	420
12.115	420
12.1525	420

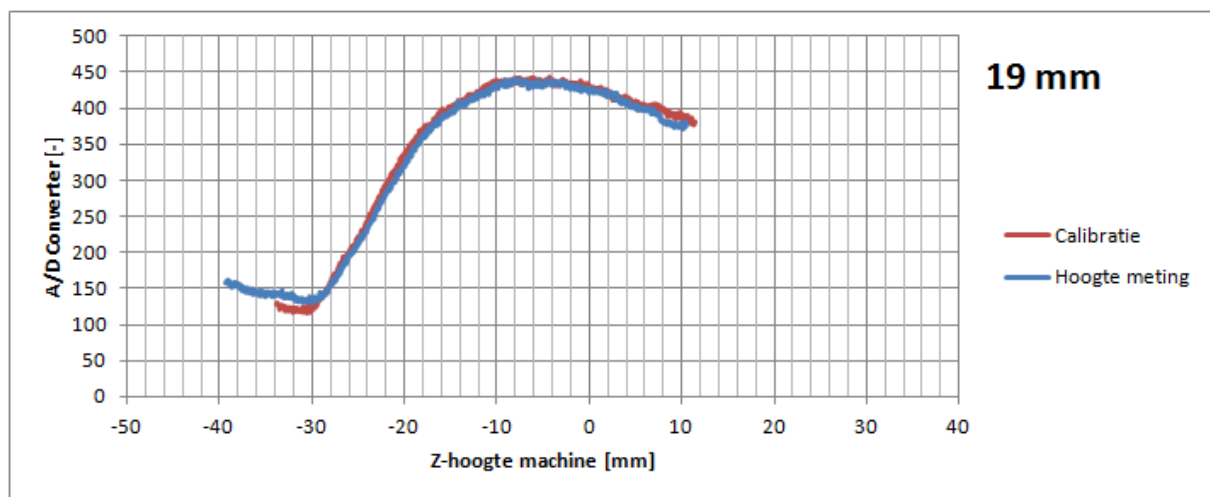
In beide gevallen is er een hoogste waarde die maar één keer voorkomt. Wanneer er dus puur gekeken wordt naar deze waarden zou betekenen dat het verschil in beide Z-posities de hoogte van het voorwerp is. Dat zou dus uitkomen op 15,4025 mm. De echte hoogte is 20,1 mm dus daar zit een afwijking in van circa 30%.

Wanneer er dus puur gekeken wordt naar de hoogste waarde zit er een grote afwijking in. Dit is ook logisch aangezien er eerder was geconstateerd dat de ruis ongeveer 5 bedraagt. Dit betekent dat alle waarden die 5 lager liggen dan de hoogste waarde ook de echte hoogste waarde kunnen zijn. Het is puur toevallig geweest dat tijdens deze meting er maar één keer een hoogste waarde uit gekomen is. Uit de tabellen kun je al zien dat alle waarden die daarin staan nog binnen de ruis zullen vallen, met andere woorden, alle Z-posities kunnen de hoogste positie zijn. Als er dan wordt gekeken naar de laagste en hoogste Z-positie komt er voor de calibratie en de hoogte meting respectievelijk een verschil uit van 3,9 mm en 4,65 mm. Dit zou dus al betekenen dat de onnauwkeurigheid van de hoogtemeting al minimaal 8,55 mm zou zijn. (de onnauwkeurigheid ligt nog hoger omdat de tabellen niet alle waarden weergeven die binnen de ruis vallen).

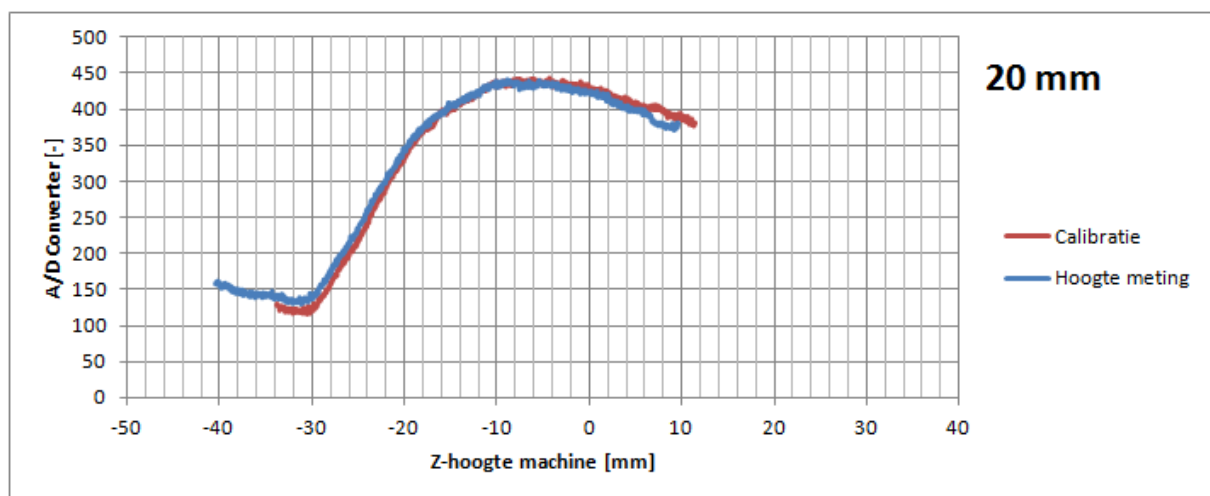
De conclusie die hieruit getrokken kan worden is dat met de huidige meetapparatuur er geen resultaat behaald kan worden wat voldoet aan de 0,1 mm nauwkeurigheid. Door de meetapparatuur te verbeteren kan het resultaat natuurlijk ook verbeterd worden. Maar er kan ook op een andere manier worden gekeken naar de meetdata en deze op een andere manier interpreteren. In plaats van te selecteren op puur de hoogste waarde is het zinvoller om alle meetdata te benutten en de twee ontstaande curven in elkaar te laten passen. Als voorbeeld is dit gedaan met Excel, de curve van de hoogtemeting is in feite een verschoven calibratie meting en ligt in dit geval ook iets lager. Om hiervoor te corrigeren is de curve met een constante factor iets omhoog gezet. Bij de waarden van Z-coördinaat is bij elke waarde een dezelfde variable toegevoegd. Op deze manier kan de hele curve verschoven worden en op het oog bepaald worden wanneer deze in overeenstemming is met de andere curve. In de volgende 4 figuren is te zien in hoeverre de curven met elkaar in overeenstemming zijn bij een correctie van 18 mm, 19 mm, 20 mm en 21 mm.



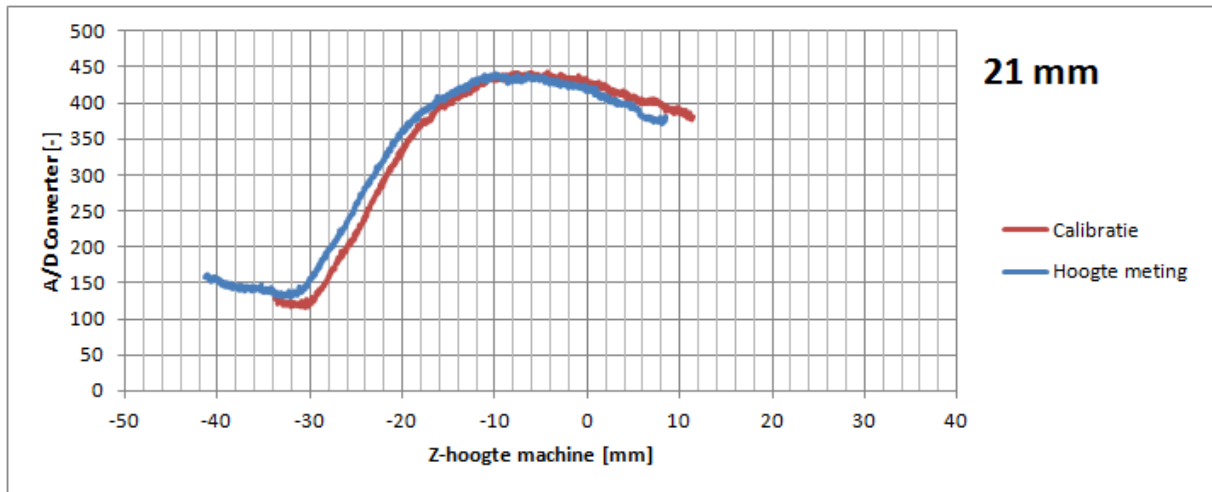
Figuur 33 Bij deze grafiek heeft de hoogte meting een verschuiving ondergaan van 18mm.



Figuur 34 Bij deze grafiek heeft de hoogte meting een verschuiving ondergaan van 19mm.



Figuur 35 Bij deze grafiek heeft de hoogte meting een verschuiving ondergaan van 20 mm.



Figuur 36 Bij deze grafiek heeft de hoogte meting een verschuiving ondergaan van 21 mm.

Op het oog is hier te zien dat de curves duidelijk beter passen bij 19 mm en 20 mm dan bij 18 mm en 21 mm. Er kan gesteld worden dat de hoogte dus tussen de 18 mm en 21 mm moet zijn. De nauwkeurigheid waarmee de hoogte bepaald kan worden is dus nu 3 mm. Dit toont aan dat door te kijken naar de volledige meetreeks een veel nauwkeuriger schatting gemaakt kan worden van de hoogte. Dit proces moet dan wel geautomatiseerd worden, er moet dus een wiskundige toepassing gemaakt worden op de meetresultaten die een best passend model moet creëren uit de meetdata.

## Hoofdstuk 6: Conclusie

De huidige opstelling voldoet niet aan alle eisen. De snelheid en nauwkeurigheid zijn niet toereikend. De snelheid wordt momenteel gelimiteerd door de USBCNC CPU. Op dat front kan er dus niet veel verbeterd worden aan de huidige opstelling.

Qua nauwkeurigheid valt er nog wel veel te winnen, dit is ingedeeld in 4 verschillende punten.

### Resolutie

De resolutie van het huidige systeem is niet goed genoeg om op 0,1 mm nauwkeurig te kunnen meten. Deze zal dus moeten worden verhoogd. Hiervoor kan er een krachtiger laser gebruikt worden met een hoger uitgangsvermogen. Het beste zou echter zijn als er een andere A/D converter gebruikt kan worden met een hoger aantal bits.

### Ruis

Door de resolutie te verhogen zal de signaal/ruis verhouding ook beter worden. Adviezen om de ruis te verminderen zijn:

- Probeer metingen te verrichten in een zo donker mogelijke ruimte.
- Gebruik afschermde kabels om de data over te brengen.
- Plaats de elektrische componenten zo ver mogelijk uit de buurt van de stappenmotoren.
- Probeer de weerstand  $R_f$  zo klein mogelijk te houden
- Gebruik elektrische componenten die zelf zo min mogelijk ruis veroorzaken, dit is vooral bedoeld bij de keuze van de OP-AMP, hierin is veel keus en verschillende de ruiswaardes enorm per type.

### Uitlijning

Om metingen af te kunnen stellen met theoretische modellen is uitlijning zeer belangrijk. Maar er gaat ook een deel van de nauwkeurigheid verloren bij slechte uitlijning van het systeem. Bij uitlijning moet er ook aan gedacht worden om de laserbundel op een zo klein mogelijk punt te focuseren.

### Data-analyse

Er moet gezocht worden naar een goede manier om de data te analyseren. Dit kan gebeuren door software te vinden die een oplossing kan bieden of anders zelf wiskunde toepassen en deze inprogrammeren.

Wanneer er met de hardware/software combinatie beschreven in dit verslag wordt gewerkt zal het meetsysteem begrenst blijven in de snelheid waarmee gemeten kan worden. De resolutie is minder effectief te verbeteren wanneer de A/D Converter van de USBCNC CPU gebruikt wordt.

Een oplossing om de resolutie toch te verbeteren en de ruis te verminderen is door een krachtiger laser te gebruiken. Qua kosten blijft een krachtiger laser vergelijkbaar. De uitlijning kan ook verbeterd worden door met een CNC-machine een behuizing te frezen waarin de laserdiode en fotodiode uitgelijnd geplaatst kunnen worden. Dit zijn twee relatief eenvoudige stappen die verbeterd kunnen worden

Een heel nieuw systeem ontwikkelen met een nieuwe A/D Converter of krachtiger software toe te passen om data-analyse uit te voeren zal veel ontwikkeling geld en tijd in gaan zitten. Er moet worden onderzocht of dat nog rendabel is.



## Bibliografie

1. **Robert F. Coughlin, Frederick F. Driscoll.** *Operational amplifiers & linear integrated circuits.* United States of America : Prentice-Hall, 1991.
2. **Zoete, Drs. Th. Mollinga m.m.v. ir. H.** *Operationele versterkers.* Overberg gem. Amerongen : Delta Press B.V., 1985.
3. **Kasap, S.O.** *Optoelectronics and Photonics; Principles and Practices.* Upper Saddle River, New Jersey 07458 : Prentice-Hall Inc., 2001.
4. *ALGaInP Visible Laser Diode ADL-65075TR/L.*

## Bijlage

<b><u>A. SOURCE CODE VAN DE MAINWINDOW</u></b> .....	FOUT! BLADWIJZER NIET GEDEFINIEERD.
<b><u>B. SOURCECODE VAN CALIBRATIE METING</u></b> .....	FOUT! BLADWIJZER NIET GEDEFINIEERD.
<b><u>C. SOURCECODE VAN XY-SCAN</u></b> .....	FOUT! BLADWIJZER NIET GEDEFINIEERD.
<b><u>D. MEETRESULTATEN VAN LASERDIODE</u></b> .....	FOUT! BLADWIJZER NIET GEDEFINIEERD.
<b><u>E. DATASHEETS</u></b> .....	FOUT! BLADWIJZER NIET GEDEFINIEERD.

## A. Source code van de mainwindow

```
#include "mainwindow.h"
#include "ui_mainwindow.h"
#include "dialog_calibrate.h"
#include "dialog_height.h"
#include "heightspeed.h"
#include "threedsan.h"
#include "speed_cali.h"
#include "XYscan.h"
#include <systemcali.h>
#include <QFileInfo>
#include <QDialog>
#include <fstream>
#include <QFile>
#include <QSound>
#include <QDesktopServices>
#include <QUrl>
#include <iostream>
#include <windows.h>
#include <xyscan.h>
#include <stdialog.h>
// Beveiliging om ervoor te zorgen dat er niet meerdere threads
// tegelijk gestart kunnen worden
bool ThreadProcessAble=true;
//schakelt een beveiliging uit
int calibrated=1;
MainWindow::MainWindow(QWidget *parent) :
    QMainWindow(parent),
    ui(new Ui::MainWindow)
{
    // Pointers gemaakt voor gebruikt van de GUI en timer
    // de timer zorgt ervoor hoevaak de GUI een refresh krijgt
    ui->setupUi(this);
    timer = new QTimer(this);
    timer->setInterval(100);
    connect(timer, SIGNAL(timeout()), this, SLOT(timer_ovf()));
    // Hier worden de threads aangemaakt
    thrXYScan = new XYScan(this);
    thrSystemCali = new SystemCali(this);
    thrMeasureHeight = new MeasureHeight(this);
    thrHeightSpeed = new heightspeed(this);
    thrScan = new threeDSan(this);
    thrSpeedCali = new speed_cali(this);
    // Dit zijn connects die ervoor zorgen dat de GUI gegevens kan
    // uitwisselen met de threads die in een ander proces plaatsvinden
    connect(this, SIGNAL(Line(double)), thrXYScan, SLOT(onLine(double)));
    connect(this, SIGNAL(NoLine(double)), thrXYScan, SLOT(onNoLine(double)));

    connect(this, SIGNAL(XYScanVelocity(double)), thrXYScan, SLOT(getVelocity(double)));

    connect(this, SIGNAL(XYScanLengthX(double)), thrXYScan, SLOT(getXlength(double)));

    connect(this, SIGNAL(XYScanLengthY(double)), thrXYScan, SLOT(getYlength(double)));

    connect(thrMeasureHeight, SIGNAL(SendHeight(double)), this, SLOT(GetHeight(double)));
}
```

```

connect(this, SIGNAL(caliStepSize(double)), thrSpeedCali, SLOT(getStepSize(double)));

connect(this, SIGNAL(caliZvalue(double)), thrSpeedCali, SLOT(getZvalue(double)));

connect(this, SIGNAL(caliSample(double)), thrSpeedCali, SLOT(getSample(double)));

connect(this, SIGNAL(Smart3DStepSize(double)), thrScan, SLOT(Scan_StepSize(double)));

connect(this, SIGNAL(Smart3DWidth(double)), thrScan, SLOT(Scan_GetWidth(double)));

connect(this, SIGNAL(Smart3DHeight(double)), thrScan, SLOT(Scan_GetHeight(double)));
    // Dit zet de afmetingen van de GUI vast
    setFixedSize(460,500);
    // Dit zorgt ervoor dat de default waarden naar de threads worden
    gestuurd
    // (kunstmatige pushbutton activatie)
    on_btnSaveCali_clicked();
    on_btnSaveSmart3D_clicked();
    on_btnSaveXYScan_clicked();
}
MainWindow::~MainWindow()
{
    // Wanneer de GUI wordt afgesloten, word het proces verwijderd
    delete ui;
}
// Wanneer er op de CONNECT pushbutton word geklikt wordt er contact
// gemaakt met de CncServer van USBCNC. Tevens stuurt deze een tekst
// naar een line-edit die de status aangeeft van de server.
// Wanneer USBCNC in simulatie draait (dus er is geen contact met
// de CPU van USBCNC uit zich dit in de GUI door van kleur te
// veranderen en met tekst "Simulation"
void MainWindow::on_btnConnect_clicked()
{
    QSound::play("ping1.wav");
    CNC_RC rc = CncConnectServer(""); // ("cnc.ini");
    ui->txtServerStatus->setText(CncGetRCText(rc));
    if (rc == CNC_RC_OK || rc == CNC_RC_ALREADY_RUNS) timer->start();
    if(CncGetSimulationMode())
    {
        this->setStyleSheet("background-color: lightgray;");
        ui->txtTEST2->setText("Simulation");
    }
}
// Hier wordt een calibratie thread aangeroepen, wanneer er al een
// thread gestart is wordt dat aangegeven. Eveneens moet de laser
// geactiveerd zijn alvorens deze thread gestart kan worden.
void MainWindow::on_btnCalibrate_clicked()
{
    if(CncGetOutput(CNC_I0ID_AUX1_OUT))
    {
        if(ThreadProcessAble)
        {
            thrSystemCali->start();
            calibrated=1;
        }
    }
}

```

```

    }
    else
    {
        ui->txtServerStatus->setText("Already running process");
    }
}
else
{
    ui->txtServerStatus->setText("Enable laser before running
calibration");
}
}
// Wanneer "Enable Laser" wordt aangevinkt zal hier de laser
// aangezet worden. Alvorens de laser aangaat zal een alarm
// geluid worden afgespeeld.
void MainWindow::on_btnAUX1_toggled(bool checked)
{
    if(checked)
    {
        QSound::play("alarm2.wav");
        Sleep(3000);
        QSound::play("LASER.wav");
        CncSetOutput(CNC_I0ID_AUX1_OUT,1);
    }
    else
    {
        CncSetOutput(CNC_I0ID_AUX1_OUT,0);
    }
}
void MainWindow::GetHeight(double value)
{
    ui->txtHeight->setText(QString::number((value), 'f',6));
}
// Binnen deze functie staan voornamelijk teksten die constant vernieuwd
// worden. Denk hierbij aan de coördinaten van de assen. Maar ook zit hier
// de beveiliging
// om te kijken of er nog een thread actief is of niet. Wanneer dat het
// geval is zal er
// geen andere thread gestart kunnen worden.
void MainWindow::timer_ovf()
{
    ui->txtX->setText(QString::number(CncGetWorkPos2(CNC_X_AXIS), 'f', 3));
    ui->txtY->setText(QString::number(CncGetWorkPos2(CNC_Y_AXIS), 'f', 3));
    ui->txtZ->setText(QString::number(CncGetWorkPos2(CNC_Z_AXIS), 'f', 3));
    ui->txtA->setText(QString::number(CncGetWorkPos2(CNC_A_AXIS), 'f', 3));
    ui->txtB->setText(QString::number(CncGetWorkPos2(CNC_B_AXIS), 'f', 3));
    ui->txtSpindle->setText(QString::number(CncGetSpindleSpeed(), 'f', 6));
    ui->txtANALOG1->setText(QString::number(CncGetInput(CNC_I0ID_ANA3_IN), 'f',6));
    ui->txtANALOG1_2->setText(QString::number(CncGetInputRaw(CNC_I0ID_ANA3_IN), 'f',6));
    if(thrScan->isRunning() || thrSpeedCali->isRunning() || thrSystemCali->isRunning() || thrMeasureHeight->isRunning() || thrXYScan->isRunning() || thrHeightSpeed->isRunning())
        {ThreadProcessAble=false;}else{ThreadProcessAble=true;}
}
// Roept een thread aan. Idem aan calibratie.
void MainWindow::on_MeasureHeight_clicked()
{
    if(calibrated!=1)
    {

```

```

        ui->txtServerStatus->setText("Please calibrate first");
    }
else
{
    if(ThreadProcessAble)
    {
        thrMeasureHeight->start();
    }
    else
    {
        ui->txtServerStatus->setText("Already running process");
    }
}
}
// thread nog in testfase
void MainWindow::on_MeasureDimensions_clicked()
{
    if(calibrated!=1)
    {
        ui->txtServerStatus->setText("Please calibrate first");
    }
    else
    {
        Dialog_Height dialog(this);
        if(dialog.exec())
        {
            CNC_POSITION cali = {0,0,0,0,0,0};
            int analog,check=0,yoffset=0,move[6] = {0,0,0,0,0,0};
            analog = 1;
            move[2]=1;
            cali.z=CncGetWorkPos2(CNC_Z_AXIS);
            while(check<30)
            {
                CncMoveTo(cali,move,1);
                Sleep(500);
                cali.z=(cali.z)+0.1;
                if(CncGetInputRaw(CNC_I0ID_ANA3_IN)>analog)
                {
                    analog=CncGetInputRaw(CNC_I0ID_ANA3_IN);
                    yoffset=cali.z;
                }
                else
                {
                    check++;
                }
            }
            cali.z=yoffset;
            CncMoveTo(cali,move,1);
        }
        std::ofstream DimensionFileX;
        DimensionFileX.open ("DimensionX_Report.txt");
        int analog=CncGetInputRaw(CNC_I0ID_ANA3_IN);
        double steptowrite=CncGetWorkPos2(CNC_X_AXIS)-100;
        double axes[CNC_MAX_AXES] = {1.0,0,0,0,0,0};
        CncStartJog(axes, 0.1, 1);
        while(CncGetInputRaw(CNC_I0ID_ANA3_IN)>(analog-100))
        {
            if(CncGetWorkPos2(CNC_X_AXIS)>steptowrite)
            {

```

```

DimensionFileX<<CncGetWorkPos2 (CNC_X_AXIS)<<"\t"<<CncGetInputRaw (CNC_I0ID_A
NA3_IN)<<std::endl;
        steptowrite=CncGetWorkPos2 (CNC_X_AXIS)+0.1;
    }
    }
    DimensionFileX.close();
    CncStopJog(0);
}

// functie om alle gaande processen te stoppen
void MainWindow::on_pushButton_4_clicked()
{
    CncStopJog(0);
}

// Hier wordt de thread aangeroepen om een XY-scan te starten.
void MainWindow::on_btnFollowLine_clicked()
{
    if(ThreadProcessAble)
    {
        thrXYScan->start();
    }
    else
    {
        ui->txtServerStatus->setText("Already running process");
    }
}

// thread nog in testfase
void MainWindow::on_btn_SmartScan_clicked()
{
    thrScan->start();
}

// thread nog in testfase
void MainWindow::on_btnSpeedCali_clicked()
{
    thrSpeedCali->start();
}

// Wanneer op de knop "Outside segment" geklinkt word
// dan stuurt deze functie informatie naar de XY-Scan thread
double MainWindow::on_btnOutsideLine_clicked()
{
    QSound::play("ping2.wav");
    Sleep(1500);
    emit NoLine(CncGetInputRaw (CNC_I0ID_ANA3_IN));
}

// Wanneer op de knop "On segment" geklinkt word
// dan stuurt deze functie informatie naar de XY-Scan thread
double MainWindow::on_btnOnLine_clicked()
{
    QSound::play("ping2.wav");
    Sleep(1500);
    emit Line(CncGetInputRaw (CNC_I0ID_ANA3_IN));
}

// thread nog in testfase
void MainWindow::on_pushButton_clicked()
{
    QDesktopServices::openUrl(QUrl("http://www.damencnc.com"));
}

// thread nog in testfase
void MainWindow::on_btnTEST_clicked()
{

```

```

    CncRunSingleLine("g92 x0 y0 z0 a0 b0");
    CNC_POSITION cali = {150,-40,30,0,0,0};
    int move[6] = {1,1,1,0,0,0};
    CncMoveTo(cali,move,0.5);
}
// Functie om alle processen te pauzeren
void MainWindow::on_btnPause_clicked()
{
    CncPauseJob();
}
// Functie op direct het contact met de server te verbreken
void MainWindow::on_btnSTOP_clicked()
{
    CncDisconnectServer();
}
// thread nog in testfase
void MainWindow::on_SpeedMeasureHeight_clicked()
{
    if(ThreadProcessAble)
    {
        thrHeightSpeed->start();
    }
    else
    {
        ui->txtServerStatus->setText("Already running process");
    }
}
// Als er gepauzeert is kan met de functie de taak
// weer hervat worden.
void MainWindow::on_btn_ResumeJob_clicked()
{
    CncRunOrResumeJob();
}
// Standaard zijn de extra instellingen niet zichtbaar. Wanneer deze
// functie wordt aangeroepen zullen de afmetingen van de GUI
// veranderen waardoor de extra instellingen beschikbaar worden.
void MainWindow::on_btnShowOptions_toggled(bool checked)
{
    if(checked)
    {
        adjustSize();
        setFixedSize(850,500);
    }
    else
    {
        adjustSize();
        setFixedSize(460,500);
    }
}
// Deze functies sturen de informatie van de ingevoerde
// waardes door naar de threads.
void MainWindow::on_btnSaveCali_clicked()
{
    emit caliStepSize(ui->CaliStepSize->value());
    emit caliZvalue(ui->CaliZdrop->value());
    emit caliSample(ui->CaliSample->value());
}
void MainWindow::on_btnSaveSmart3D_clicked()
{
    emit Smart3DStepSize(ui->Smart3DStepSize->value());
    emit Smart3DWidth(ui->Smart3DWidth->value());
}

```



```

        emit Smart3DHeight(ui->Smart3DHeight->value());
    }
void MainWindow::on_btnSaveXYScan_clicked()
{
    emit XYScanVelocity(ui->XYScan_Velocity->value());
    emit XYScanLengthX(ui->XYScan_Xlength->value());
    emit XYScanLengthY(ui->XYScan_Ylength->value());
}
// Deze functies zetten de waarden van de
// instellingen terug naar het begin.
void MainWindow::on_btnDefaultCali_clicked()
{
    ui->CaliStepSize->setValue(0.1);
    ui->CaliZdrop->setValue(5);
    ui->CaliSample->setValue(1000);
}
void MainWindow::on_btnDefaultSmart3D_clicked()
{
    ui->Smart3DStepSize->setValue(0.1);
    ui->Smart3DWidth->setValue(10);
    ui->Smart3DHeight->setValue(10);
}
void MainWindow::on_btnDefaultXYScan_clicked()
{
    ui->XYScan_Velocity->setValue(0.1);
    ui->XYScan_Xlength->setValue(5);
    ui->XYScan_Ylength->setValue(1000);
}
// Deze functies openen het laatst aangemaakte bestand
// van de desbetreffende functie in een nieuw venster.
void MainWindow::on_btn_OpenCaliFile_clicked()
{
    system("START SpeedCali_Report.txt");
}
void MainWindow::on_btn_OpenSmart3D_clicked()
{
    system("START XYZScan.txt");
}
void MainWindow::on_btn_OpenXYScan_clicked()
{
    system("START XYScan_Report.txt");
}

```

## B. Sourcecode van calibratie meting

```
#include "speed_cali.h"
#include "cncapi.h"
#include <fstream>
#include <iostream>
#include <QtCore>
speed_cali::speed_cali(QObject *parent) :
    QThread(parent)
{
}
// Vanuit onderstaande 3 functies haalt deze thread de gegevens die zijn
// ingevoerd in de GUI binnen
// Hiermee wordt aangegeven om de hoeveel ms er een sample moet worden
// genomen
void speed_cali::getSample(double value)
{
    Sample=value;
}
// Hiermee kan worden aangegeven met welke snelheid de Z-as zicht beweegt
void speed_cali::getStepSize(double value)
{
    Velocity=value;
}
// Hiermee wordt aangegeven hoever de Z-as omhoog moet bewegen
void speed_cali::getZvalue(double value)
{
    HeightUp=value;
}
void speed_cali::run()
{
    // De CNC_POSITION struct wordt hier aangemaakt onder de naam "Scan" en
    // alle assen worden op 0 gezet
    CNC_POSITION Scan = {0,0,0,0,0,0};
    // Een array om aan te geven welke assen moeten gaan bewegen wordt hier
    // aangemaakt en allemaal op 0 gezet
    int move[6] = {0,0,0,0,0,0};
    // Twee variable wordt aangemaakt om te kunnen bepalen waar het hoogste
    // punt zit
    float Height;
    float Highest=-1;
    // Met deze functie wordt de huidige positie op de CNC-besturing
    // genult. Dit houdt in dat er vanaf nu wordt
    // gekeken naar het werkstukcoördinatenstelsel van de CNC-machine
    CncRunSingleLine("g92 x0 y0 z0");
    // De derde as uit de move array wordt op 1 gezet, dit betekent dat de
    // Z-as nu kan gaan bewegen wanneer de
    // functie CncMoveTo() wordt aangeroepen.
    move[2]=1;
    // Het coördinaat waar de Z-as naartoe moet bewegen wordt hier bepaald.
    // De hoogte die omhoog moet worden
    // bewogen wordt opgeteld bij de huidige positie.
    Scan.z=CncGetMachinePos2(CNC_Z_AXIS)+HeightUp;
    // functie nodig om de library te openen om bestanden te kunnen
    // aanmaken
    std::ofstream SpeedCali;
    // bestand wordt aangemaakt waarnaar gegevens zullen weggeschreven
    // worden
    SpeedCali.open ("SpeedCali_Report.txt");
    // De Z-as zal nu in gaan bewegen naar de ingestelde positie met de
    // ingestelde snelheid.
```

```

CncMoveTo(Scan,move,Velocity);
// Terwijl de Z-as beweegt, gaat het programma gewoon verder, om die
reden zal deze while-loop dus gestart
// worden terwijl de Z-as begint te bewegen. Wanneer de juiste Z-hoogte
is bereikt stopt de While loop.
while(CncGetWorkPos2(CNC_Z_AXIS)<HeightUp)
{
    // Hier wordt de coördinaten van de machine en van de
werkpositie weggeschreven naar een bestand
    // samen met de waarde van de analoge ingang

SpeedCali<<CncGetMachinePos2(CNC_Z_AXIS)<<"\t"<<CncGetWorkPos2(CNC_Z_AXIS)<
<"\t"<<CncGetInputRaw(CNC_I0ID_ANA3_IN)<<std::endl;
    // Dit gedeelte is om te kijken wat de hoogste waarde van
de analog ingang was
    if(CncGetInputRaw(CNC_I0ID_ANA3_IN)>Highest)
    {
        Highest=CncGetInputRaw(CNC_I0ID_ANA3_IN);
        Height=CncGetWorkPos2(CNC_Z_AXIS);
    }
    // Hierbij kan worden ingesteld hoelang het systeem nu
vervolgens moet wachten
    // met de While loop opnieuw uit te voeren
    msleep(Sample);
}

// Hier wordt onderaan het bestand gezet wat de begin instellingen
waren

SpeedCali<<"Velocity"<<"\t"<<Velocity<<std::endl<<"Wait"<<"\t"<<Sample<<std
::endl<<std::endl;
    // Hier wordt onderaan het bestand gezet bij welke hoogte analog input
het hoogst was
SpeedCali<<std::endl<<"Height"<<"\t"<<Height;
    // Sluit het bestand
SpeedCali.close();
    // Het bestand wordt nu automatisch geopent in een nieuw venster
system("START SpeedCali_Report.txt");
}

```

### C. sourcecode van XY-Scan

```
#include "xyscan.h"
#include "cncapi.h"
#include "stddialog.h"
#include <mainwindow.h>
#include <fstream>
#include <iostream>
#include <QtCore>
#include <QDialog>
XYScan::XYScan(QObject *parent) :
    QThread(parent)
{
}
// Inlezen welke waarde de Analoge Input heeft van USBCNC voor
// op een segment en erbuiten
void XYScan::onLine(double value)
{
    Line=value;
}
void XYScan::onNoLine(double value)
{
    NoLine=value;
}
// Instellingen uit de GUI worden hiermee ingelezen
void XYScan::getVelocity(double value)
{
    Velocity=value;
}
void XYScan::getXlength(double value)
{
    LengthX=value;
}
void XYScan::getYlength(double value)
{
    LengthY=value;
}
// Enkele instellingen die in (nu nog) in de source code zitten
void XYScan::run()
{
    //----Fill in----//
    float StepSize=0.5;
    float wait=50;    // dit bepaald de sample rate
    bool Gcode=false; // functie om een Gcode te genereren, beta
    //----Fill in----//
    // Variabelen aanmaken om de besturing te kunnen aansturen
    double offset;
    CNC_POSITION Scan = {0,0,0,0,0,0};
    int move[6] = {0,0,0,0,0,0};
    // De huidige positie wordt weggeschreven naar variabelen
    Scan.x=CncGetMachinePos2(CNC_X_AXIS);
    Scan.y=CncGetMachinePos2(CNC_Y_AXIS);
    // Geeft aan dat de X en Y aangestuurd mogen worden
    move[0]=1;move[1]=1;
    // Emprische bepaalde formule om te bepalen wanneer er iets moet worden
    // weggeschreven
    offset=((NoLine-Line)/2)+Line;
    // Onderstaande code kan ervoor zorgen dat de scanner constant meet
    // gebruikt voor testwerk
    //-----overrule the offset-----//
    offset=1024;
```

```

        wait=50;
//-----overrule the offset-----//
// De assen worden genult, nodig voor het aansturen
    CncRunSingleLine("g92 x0 y0 z0");
// bestanden worden aangemaakt en geopent
// Gcode wordt alleen geactiveerd als deze wordt aangezet
// in de source code
    std::ofstream XYScanReport;
    std::ofstream XYScanGcode;
    XYScanReport.open ("XYScan_Report.txt");
    if(Gcode){XYScanGcode.open ("XYScanGcode.txt");}
// Er wordt een while loop gestart waar pas wordt uitgestapt op het
// moment dat de ingestelde Y-waarde is bereikt.
    while(CncGetWorkPos2(CNC_Y_AXIS)<LengthY)
    {
        // De machine wordt naar de ingestelde x-positie gestuurd
        // er wordt een while-loop gestart waarin steeds wordt gekeken
        // naar de offset waarde en de analoge ingang of er iets moet
gebeuren
        // dit blijft zich door de while loop net zo lang herhalen
totdat de
        // x-positie is bereikt.
        Scan.x=(CncGetMachinePos2(CNC_X_AXIS)+LengthX);
        CncMoveTo(Scan,move,Velocity);
        while(CncGetWorkPos2(CNC_X_AXIS)<LengthX)
        {
            if(CncGetInputRaw(CNC_I0ID_ANA3_IN)<offset)
            {
XYScanReport<<CncGetWorkPos2(CNC_X_AXIS)<<"\t"<<CncGetWorkPos2(CNC_Y_AXIS)<
<<"\t"<<CncGetInputRaw(CNC_I0ID_ANA3_IN)<<std::endl;
                if(Gcode){XYScanGcode<<"g0"<<"
"<<"x"<<CncGetWorkPos2(CNC_X_AXIS)<<"
"<<"y"<<CncGetWorkPos2(CNC_Y_AXIS)<<std::endl;}
                msleep(wait);
            }
        }
        // Hier wordt de machine aangestuurd net als in voorgaande
voorbeeld, alleen
        // gaat het nu om de Y-as
        Scan.y=(CncGetMachinePos2(CNC_Y_AXIS)+StepSize);
        CncMoveTo(Scan,move,Velocity);
        while(CncGetWorkPos2(CNC_Y_AXIS)<(StepSize))
        {
            if(CncGetInputRaw(CNC_I0ID_ANA3_IN)<offset)
            {
XYScanReport<<CncGetWorkPos2(CNC_X_AXIS)<<"\t"<<CncGetWorkPos2(CNC_Y_AXIS)<
<<std::endl;
                if(Gcode){XYScanGcode<<"g0"<<"
"<<"x"<<CncGetWorkPos2(CNC_X_AXIS)<<"
"<<"y"<<CncGetWorkPos2(CNC_Y_AXIS)<<std::endl;}
                msleep(wait);
            }
        }
        // Hier wordt de machine terug gestuurd naar de begin positie
op de X-as
        // dit gebeurt op 100% snelheid en er wordt niks gemeten
ondertussen
        Scan.x=(CncGetMachinePos2(CNC_X_AXIS)-LengthX);
        CncMoveTo(Scan,move,1);

```

```

while (CncGetWorkPos2 (CNC_X_AXIS)>0)
{
}
// Zelfde principe als eerste 2 bewegingen
Scan.y=(CncGetMachinePos2 (CNC_Y_AXIS)+StepSize);
CncMoveTo (Scan,move,Velocity);
while (CncGetWorkPos2 (CNC_Y_AXIS)<(StepSize))
{
    if (CncGetInputRaw (CNC_I0ID_ANA3_IN)>offset)
    {
XYScanReport<<CncGetWorkPos2 (CNC_X_AXIS)<<"\t"<<CncGetWorkPos2 (CNC_Y_AXIS)<
<std::endl;
        if (Gcode) {XYScanGcode<<"g0"<<"
"<<"x"<<CncGetWorkPos2 (CNC_X_AXIS)<<"
"<<"y"<<CncGetWorkPos2 (CNC_Y_AXIS)<<std::endl;}
        msleep (wait);
    }
}
}
XYScanGcode<<Line<<"\n"<<NoLine;
XYScanReport.close();
if (Gcode) {XYScanGcode.close();}
system("XYScan_Report.txt");
}

```

## D. Rekenmodel

```
#include <QtCore/QCoreApplication>
#include <math.h>
#include <iostream>
#include <complex>
#include <windows.h>
#include <fstream>
#include <QFileInfo>
using namespace std;
/*hoeveel mW per vierkante centimeter*/
double GetmWM(double vermogen, double afstand, double verlies)
{
    float mWperCM=(vermogen/((2*3.1415*pow(afstand,2)))*verlies;
    return(mWperCM);
}
/*hoeveel mW er op de lens valt*/
double GetmWLens(double vermogen, double r, double verlies, double
verlies2, double verlies3)
{
    float Pfoto=vermogen*(3.1415*pow(r,2))*(verlies*verlies2*verlies3);
    return(Pfoto);
}
/*welke stroom er wordt opgewekt in de fotodiode*/
double GetAW(double hoek)
{
    double AW;
    hoek=abs(hoek);
    if(hoek>50){AW=0;}
    if(hoek==50 || hoek<50 && hoek>40){AW=((50-hoek)*0.001);}
    if(hoek==40 || hoek<40 && hoek>20){AW=((40-hoek)*0.0035)+0.01;}
    if(hoek==20 || hoek<20 && hoek>15){AW=((20-hoek)*0.024)+0.08;}
    if(hoek==15 || hoek<15 && hoek>5){AW=((15-hoek)*0.075)+0.2;}
    if(hoek==5 || hoek<5){AW=((5-hoek)*0.01)+0.95;}
    AW=AW*0.62*0.72;
    return(AW);
}
/*van radialen naar graden*/
double Deg(double D)
{
    float PI;
    PI=3.14159265358979323846;
    D=D*(180/PI);
    return(D);
}
/*Hoek ten opzichte van de y-as naar foto-diode*/
double GetAngle(double h, double xld)
{
    return((atan(xld/h)));
}
/* Hier start het programma*/
int main(int argc, char *argv[])
{
    QCoreApplication a(argc, argv);
    double
    GetmWM(double,double,double),GetmWLens(double,double,double,double,double),
    GetI(double,double),GetV(double),GetAW(double),GetAngle(double,double),Deg(
double);
    float
    Plaser,x,Dlens,Pfoto,mWpercm,Rlens,AWfoto,Ifoto,VerliesLensL,VerliesLensF,V
erliesLucht,VerliesLambert,Verlies,Versterking,Xld,corhoek;
```

```

VerliesLensL=1;
VerliesLensF=1;
VerliesLucht=1;
VerliesLambert=1;
/*Verlies op 0 betekent geen verlies meenemen in berekening*/
Verlies=0;
if(Verlies==1)
{
VerliesLensL=0.95;
VerliesLensF=0.8;
VerliesLucht=0.5;
VerliesLambert=0.90;
}
//Vermogen Laser (mW)//
Plaser=0.6;
//Afstand (cm)//
x=9;
//Diameter Lens (cm)//
Dlens=0.52;
//Versterking OP-AMP//
Versterking=5000000;
//Afstand tussen laser en diode (cm)//
Xld=1.5;
corhoek=Deg(atan(Xld/x));
Rlens=Dlens/2;
ofstream myfile;
myfile.open ("Lens2.txt");
myfile<<"Instelde data:"<<endl<<endl;
myfile<<"Vermogen laser"<<"\t"<<Plaser<<"mW"<<endl;
myfile<<"Hoek diode"<<"\t"<<corhoek<<" graden"<<endl;
myfile<<"Diameter diode-lens"<<"\t"<<Dlens<<" cm"<<endl;
myfile<<"Weerstands-waarde op-amp"<<"\t"<<Versterking<<" Ohm"<<endl;
myfile<<"Afstand laser en diode"<<"\t"<<Xld<<" cm"<<endl<<endl;
myfile<<"Verliezen:"<<endl<<endl;
myfile<<"Verlies lens bij laser"<<"\t"<<VerliesLensL<<endl;
myfile<<"Verlies in de lucht"<<"\t"<<VerliesLucht<<endl;
myfile<<"Verlies door niet-Lambertse
straler"<<"\t"<<VerliesLambert<<endl<<endl<<endl;

myfile<<"x"<<"\t"<<"MWperM"<<"\t"<<"MWopLENS"<<"\t"<<"Ifoto"<<"\t"<<"V"<<"\t"
<<"hoek"<<"\t"<<"hoek op diode"<<"\t"<<"A/W"<<std::endl;
x=0;
while(x<18)
{
// Afstand in mm
myfile<<x*10<<"\t";
// Waarde van de AD-Converter

myfile<<(((GetmWLens(GetmWM(Plaser,x,VerliesLensL),Rlens,VerliesLensF,VerliesLucht,VerliesLambert)*GetAW(-corhoek+Deg(GetAngle(x,Xld))))/1000)*Versterking)/(3.3/1023)<<"\t";
// mW per m2 van oppervlakt bol
myfile<<GetmWM(Plaser,x,VerliesLensL)<<"\t";
// mW wat op de lens van de fotodiode terecht komt

myfile<<GetmWLens(GetmWM(Plaser,x,VerliesLensL),Rlens,VerliesLensF,VerliesLucht,VerliesLambert)<<"\t";
// Stroom door de fotodiode

myfile<<(GetmWLens(GetmWM(Plaser,x,VerliesLensL),Rlens,VerliesLensF,VerliesLucht,VerliesLambert)*GetAW(-corhoek+Deg(GetAngle(x,Xld))))/1000<<"\t";

```



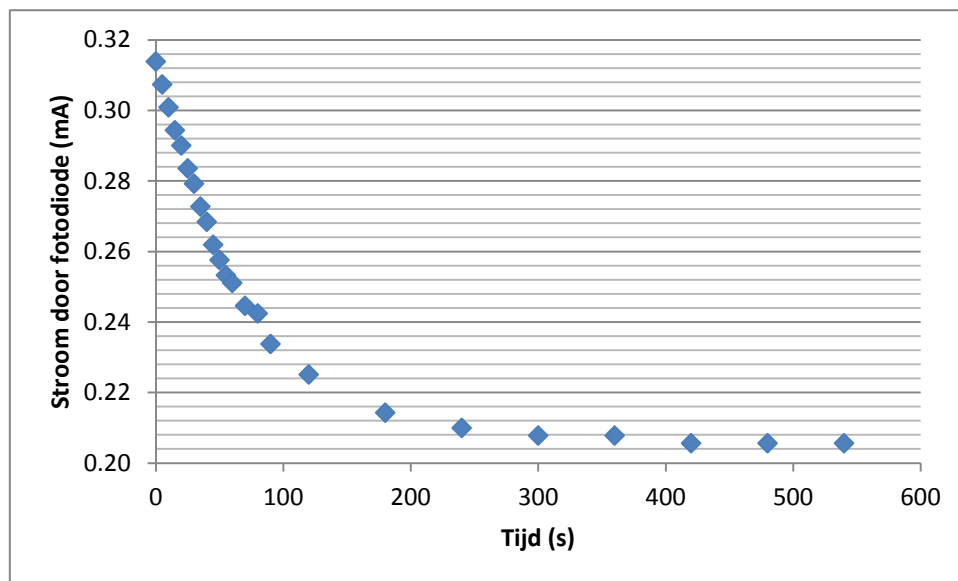
```

        // Voltage uit de OP-AMP

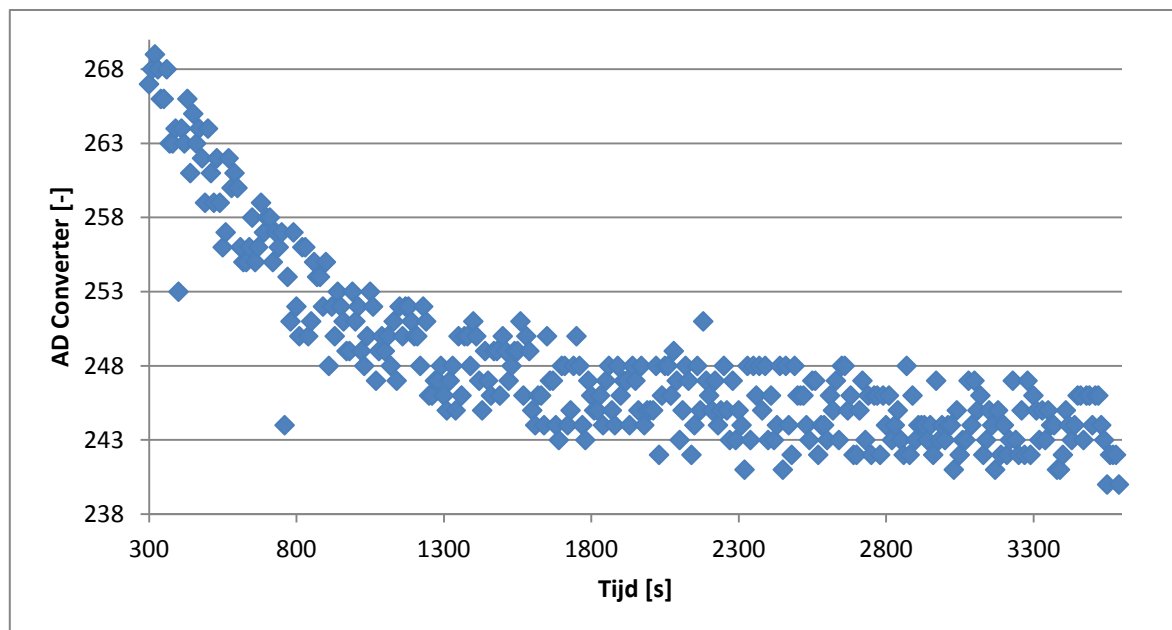
myfile<<(((GetmWLens(GetmWM(Plaser,x,VerliesLensL),Rlens,VerliesLensF,VerliesLucht,VerliesLambert)*GetAW(-corhoek+Deg(GetAngle(x,Xld))))/1000)*Versterking)<<"\t";
    // Hoek met oppervlak
myfile<<Deg(GetAngle(x,Xld))<<"\t";
    // Hoek waarin het licht de fotodiode binnenkomt
myfile<<-corhoek+Deg(GetAngle(x,Xld))<<"\t";
    // De totale AW-waarde met correctie voor de hoek
myfile<<GetAW(-corhoek+Deg(GetAngle(x,Xld)))<<endl;
    x=x+0.01;
}
myfile.close();
system("START Lens2.txt");
cout<<"Done";
return a.exec();
}

```

### E. Meetresultaten van laserdiode ADL-65075TR/L



Grafiek toont de verhouding tussen het uitgangsvermogen van de laserdiode en de tijd. De stroom door de fotodiode is rechtevenredig met het uitgangsvermogen.



Het gedrag van het systeem wanneer deze kijkt naar het vermogen dat valt op de fotodiode. Het vermogen wat op de fotodiode valt wordt omgezet in een spanning en deze wordt geconverteerd door de AD Converter. De LSB waarde van de converter is 32mV.

## *F. Datasheets*