

Industriële wetenschappen

“De polyp”

Mentoren

Wout Crevits

D. Vansteenlandt

Robbe Ryckeboer

K. Werbrouck

Corneel van der Linde

K. Geeraert

D. Goethals

T. Vandenbulcke

DOSSIER GEINTEGREERDE PROEF 2022-2023

VRIJ TECHNISCHE INSTITUUT St. Aloysius | Papebrugstraat 8A | 8820 TORHOUT

e-mail: vti@sint-rembert.be | website: <http://www.sint-rembert.be/vtiq>

1. Voorwoord

In het 4de, 5de en 6de jaar zien we veel moeilijke leerstof waarin mechanica, elektriciteit, elektronica en wiskunde aan bod komen. Omdat deze leerstof vaak moeilijk te visualiseren valt, vonden wij het een goed idee om hieraan praktische voorbeelden te koppelen, dit doen we aan de hand van de polyp. Hiermee kan vooral de leerstof mechanica in het 4de jaar veel gemakkelijker aangebracht worden zodat de leerlingen de leerstof sneller en beter kunnen begrijpen.

Inhoudstafel

1.	Voorwoord	2
2.	Inleiding	6
3.	Mechanische studie	7
3.1	Ontwerp	7
3.1.1	Brainstormen	7
3.1.2	Uiteindelijk ontwerp	8
3.2	3D-printer	12
3.2.1	Geschiedenis	12
3.2.2	Methoden	12
3.2.2.1	Methode 1: Fijn poeder	12
3.2.2.2	Methode 2: vloeistoffen	13
3.3	Materialen	13
3.3.1	Filament	13
3.3.1.1	PLA	13
3.3.2	MDF	14
Voordelen		14
Nadelen		14
3.3.3	PC-plaat	15
Voordelen		15
Nadelen		15
3.3.4	Easy systems profielen	15
Voordelen		15
Nadelen		16
4.	Elektrische studie	16
4.1	Pasco-sensoren	16
4.1.1	Krachtsensor	16
4.1.2	Rotatiesensor	17
4.2	Motor	18
4.3	Sleepcontacten	21

4.3.1	Constructie van de sleepcontacten	22
4.4	Batterij	23
4.4.1	Praktijk/gebruik	23
4.4.2.1	Lithium-ion-accu	24
4.4.2.2	Battery shield	24
5.	Elektronische studie	25
5.1	Arduino	25
5.2	ESC	26
5.3	NRF 24	28
5.4	Shields	30
5.5	“Servo”	31
5.5.1	Algemeen	31
5.5.2	Hoeksnelheid van de servomotor	32
6.	Programma’s	35
6.1	Transmitter	35
6.2	Receiver	38
7.	Wetenschappelijke studie	42
7.1	Meten van krachten	42
7.1.1	Theorie	42
7.1.1.1	Centripetale of middelpuntzoekende kracht	42
7.1.1.2	Centrifugale of middelpuntvliedende kracht	43
7.1.2	Praktijk	44
7.1.2.1	Meetopstelling met karretjes	44
7.1.2.1.1	Meetresultaten	45
7.2	Meten van rotatiesnelheid	50
7.2.1	Theorie	50
7.2.1.1	Eenparig cirkelvormige beweging (ECB)	50
7.2.1.2	Eenparig veranderlijke cirkelvormige beweging (EVCB)	51
7.2.2	Praktijk	52
7.3	Cycloïden	57
7.3.1	Hypocycloïde	58

7.3.2	Epicycloïden	62
7.3.3	Cardioïden	66
	Veiligheid	68
8.	ICT-studie	68
8.1	Website	68
9.	Evaluatie – besluit	70

2. Inleiding

De polyp is een kermisattractie die bestaat uit vier tot acht armen, aan deze armen zijn bakjes bevestigd waarin de personen kunnen zitten.

In de polyp draaien die armen en bakjes elk rond een eigen as, we zullen dus een dubbele rotatiebeweging bekomen.

In onze GIP is het de bedoeling dat we zo'n polyp zullen nabouwen in het klein. Aan de hand van deze opstelling zullen we wetenschappelijke studies uitvoeren, zo kunnen we zaken aantonen zoals de middelpuntvliedende en middelpuntzoekende kracht. Ook zullen we door de dubbele rotatiebeweging allerlei krommen kunnen bekomen.

De uiteindelijke bedoeling is dat deze opstelling in de volgende jaren kan gebruikt worden om deze zaken praktisch voor te stellen aan de jeugd, en dat ze hiermee ook eens kunnen experimenteren.

3. Mechanische studie

3.1 Ontwerp

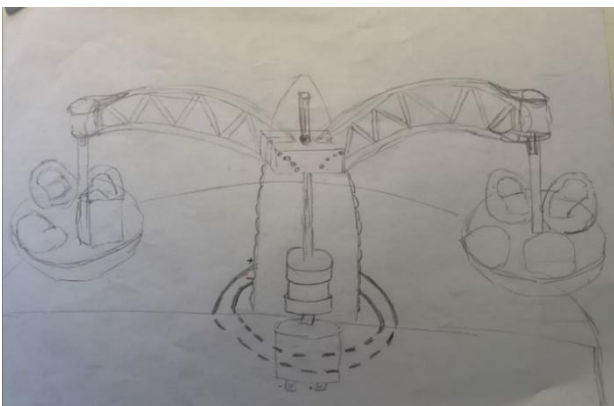
3.1.1 Brainstormen

In het ontwerp is natuurlijk enkele maanden tijd ingekropen, het heeft dan ook even geduurd om hetgeen te bekomen wat we nu hebben.

Het idee was dus om een polyp te construeren, het was de bedoeling om aan de hand van deze polyp verschillende wetenschappelijke experimenten uit te voeren.

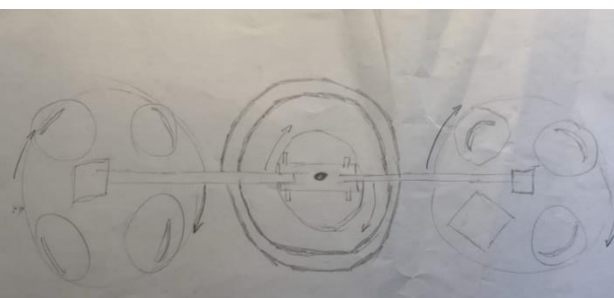
De polyp moest dan ook aan enkele belangrijke voorwaarden of specificaties voldoen.

- Hij moest kunnen cycloïden vormen, twee cirkelvormige bewegingen waren dan ook noodzakelijk.
- Er moest een opstelling op gezet kunnen worden om middelpuntvliedende kracht aan te tonen en dus te kunnen meten.
- Het moest een slank en zo compact mogelijk design worden.
- In het aandrijvingssysteem wilden we zo weinig mogelijk met tandwielen werken (dit was de boosdoener in een GIP van vorig jaar, waar onze polyp op gebaseerd is).
- Een ontwerp waarbij de inwendige werking zoveel mogelijk visueel gemaakt wordt.
- Een sterke en robuuste opstelling die tegen een stoot kan.
- Elektriciteit die via sleepcontacten overgebracht wordt.



Figuur 1 : schets 1

Eerste ontwerp, in het midden zien we de DC-motor. Aan de arm hangen nog 'bakjes' met stoeltjes erop. Het is duidelijk dat we hier nog veel meer van het idee uitgingen dat we een echte polyp gingen bouwen. We bekeken het nog niet uit de wetenschappelijke hoek, en hielden nog geen rekening met de specificaties die hierboven opgesomd zijn. Al deze specificaties komen tot hun recht in het tweede ontwerp.



Figuur 2 : schets 2

Dit is het bovenaanzicht van ons eerste ontwerp. Wat wel opvalt, is dat we vanaf moment één al wisten dat we met sleepcontacten gingen werken.

3.1.2 Uiteindelijk ontwerp

In het uiteindelijke ontwerp hebben we zoals hierboven al verteld geen gebruik gemaakt van tandwielen, hierdoor konden we een veel compactere onderkamer creëren.

1) Bevestiging motor

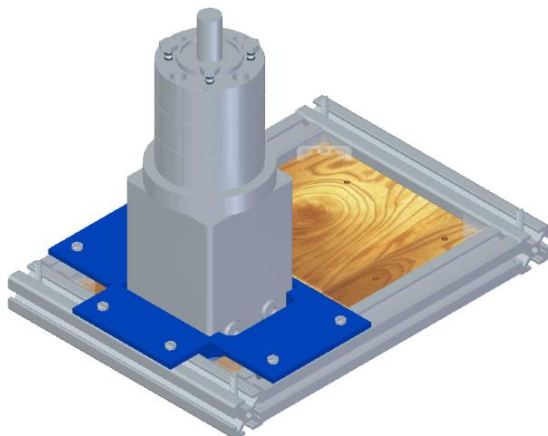
De motor is het middelpunt van onze opstelling, de gehele polyp is hierrond geconstrueerd.

Onze motor heeft alleen maar vier bevestigingspunten aan de bovenkant, hiermee hebben we dan ook gewerkt en zo hebben we de bovenplaat hieraan bevestigd. Ook staat de motor op onze grondplaat en zit deze in een bakje tegen torsie. Hieruit is het rechtstreekse gevolg dat de hoogte van de onderkamer en dus ook van de profielen gelinkt is aan de hoogte van de motor.



Figuur 3 : Bevestiging motor

2) De onderkamer

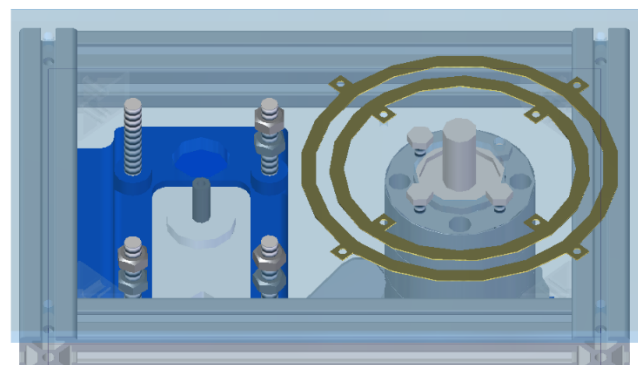


Figuur 4 : onderkamer

Om de breedte en de lengte van onze onderkamer te bepalen hebben we eerst alle onderdelen in een Assembly geplaatst, dit is een onderdeel van Solid Edge waarin je verschillende parts kunt plaatsen. We zetten alle onderdelen zo dicht mogelijk bij elkaar en hierrond tekenden we de profielen. De profielen zijn van het type Rexroth, dit zijn zeer gebruiksvriendelijke en sterke profielen. In de onderkamer bevinden zich de motor, de eerste Arduino (deze dient om de hoofdmotor aan te sturen) en de rotatiesensor. De grondplaat bestaat uit MDF en is gelaserd met de lasercutter.

3) Bovenplaat

Ook is op de profielen onze bovenplaat bevestigd, deze bestaat uit plexiglas. Dit was één van de voorwaarden waaraan de polyp moest voldoen. Aan de bovenplaat hangt de rotatiesensor, deze sensor zit in een houder. Op de bovenplaat zijn ook de sleepringen bevestigd, deze zijn samen met de bovenplaat gefreesd (meer info bij puntje 5.2 sleepringen).



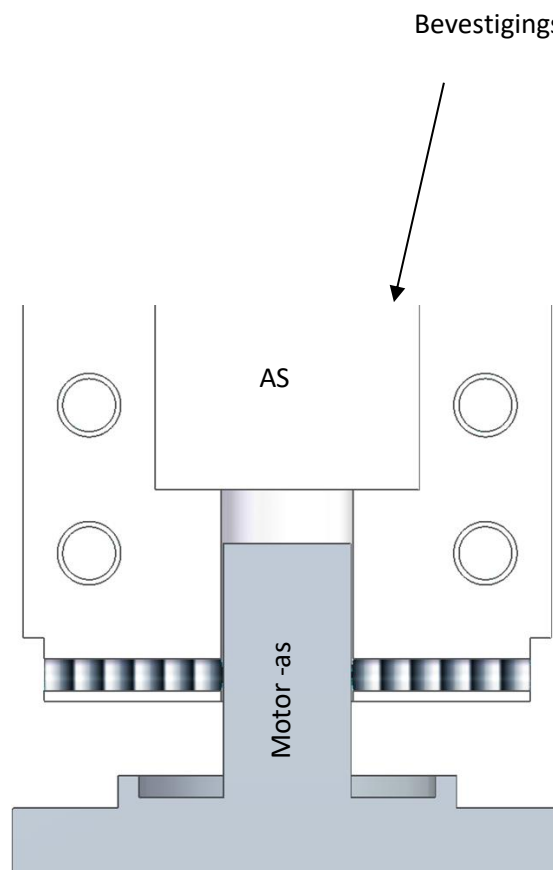
Figuur 5 : bovenplaat

4) Klemstuk

Om de as aan de hoofdmotor te bevestigen maakten we hiervoor een bevestigingsstuk. Dit bevestigingsstuk is gemaakt uit aluminium in de metaalbewerking. De bedoeling was dat we uit een cilindervormig stuk boringen maakten. Van deze boringen zijn er twee verticaal geboord. De eerste boring (kleinste) is met een even grote diameter geboord als die van de as van de motor, de tweede boring (grootste) is geboord met een even grote diameter als de diameter van de as waarop de opstellingen zullen gebouwd worden. Als deze boringen gemaakt waren, hebben we het gehele stuk in tweeën gezaagd, hierdoor is dus materiaal weggehaald van de twee boringen en zo bekomen we dus het klemmeffect. Ook hebben we horizontale boringen gemaakt, deze zullen dienen om het stuk aan te spannen. De boringen onderaan het stuk dienen voor de bevestiging van de sleepcontacten.

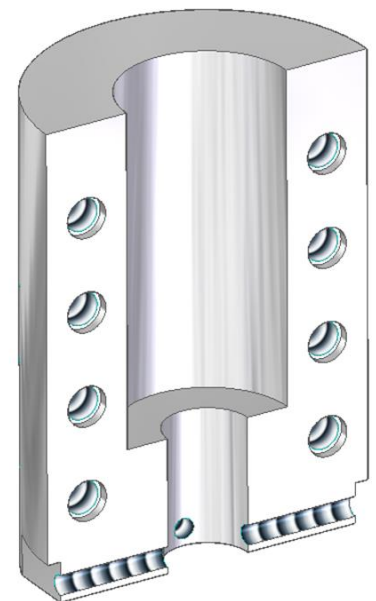


Figuur 6 : as, motor, bevestigingsstuk



Figuur 7 : doorsnede bevestiging

Bevestigingsstuk

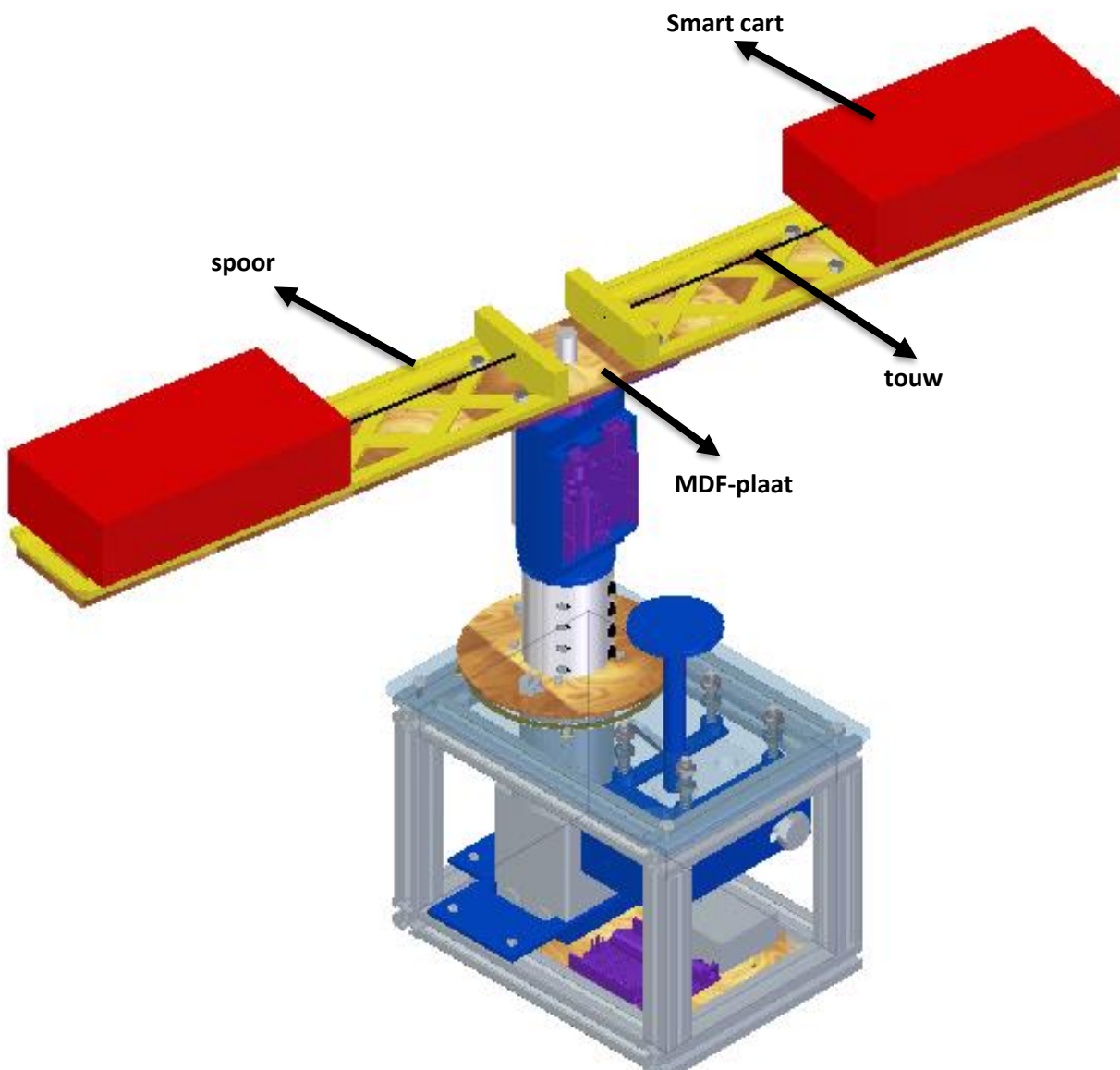


Figuur 8 : doorsnede bevestigingsstuk

5) Opstelling 1

Toen we een as hadden die op dezelfde rotatiesnelheid draaide als die van de motor konden we beginnen met de opstellingen te construeren die op deze as moesten komen.

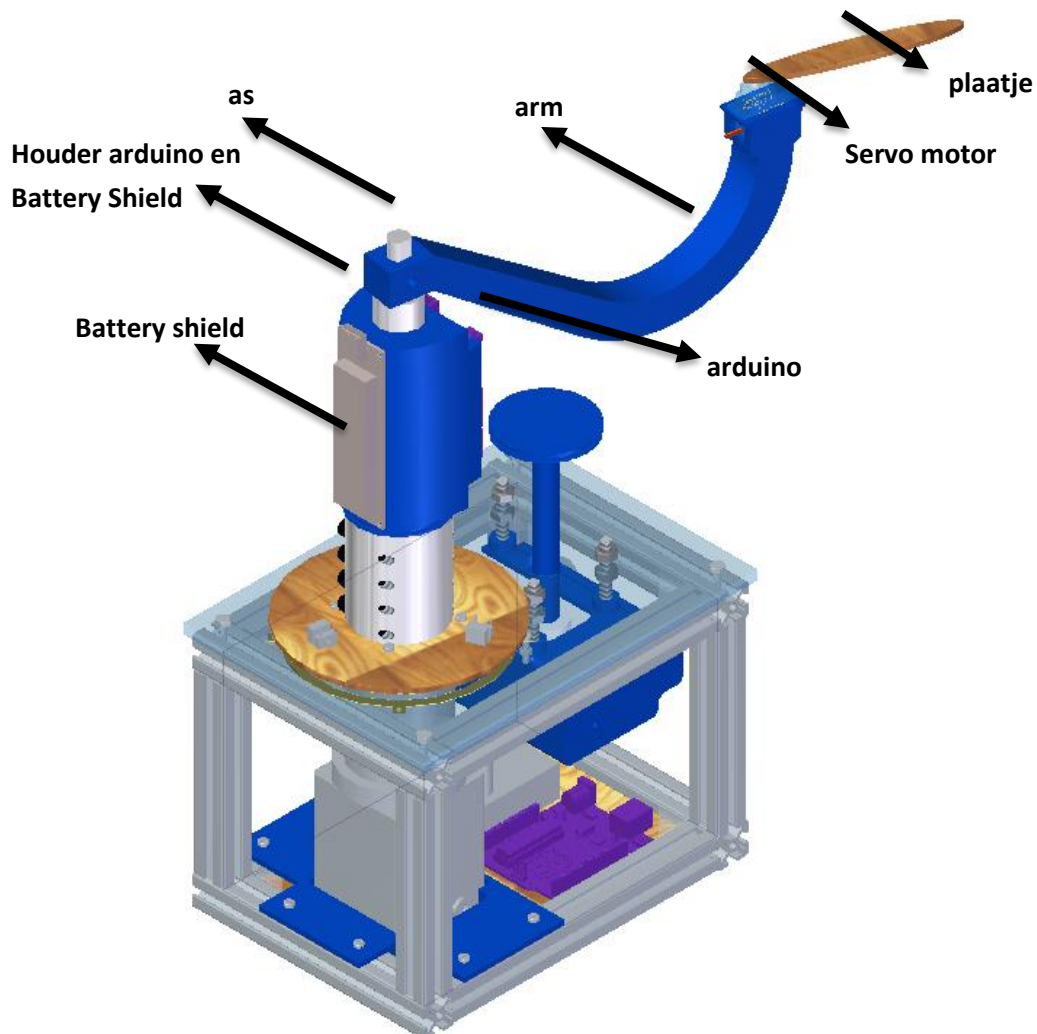
De eerste opstelling die we maakten, was deze om de krachtmeting mee uit te voeren. We gebruikten de PASCO-krachtkarretjes, deze karretjes kunnen rijden op sporen die hiervoor speciaal ontworpen zijn (deze sporen zijn vorig jaar geconstrueerd voor de looping). De sporen hebben we vastgemaakt op een lange MDF-plaat van zes mm dik. Op de sporen die het dichtst bij de arm zitten, hebben we ook een stuk gemaakt waaraan we de touwtjes kunnen bevestigen, we zorgden ervoor dat dit bevestigingsstuk zich op dezelfde hoogte bevond als de krachtsensor. Dit omdat het touw horizontaal zou hangen, zo konden de krachtmetingen op een zo correct mogelijke manier verlopen.



Figuur 9 : opstelling krachtmeting

6) Opstelling 2

De tweede opstelling is deze om de cycloïden te bepalen, hiervoor hebben we een tweede roterende beweging nodig dus ook een tweede motor. Om die tweede motor aan te drijven maken we gebruik van de sleepcontacten (zie 5.2). Ook hebben we om de motor aan te sturen en te voeden een arduino en batterij nodig op het ronddraaiende gedeelte. Omdat dit nogal grote onderdelen zijn om te bevestigen hebben we deze verticaal rond de as geplaatst. We tekenden hiervoor houders, de houder van de arduino werkt aan de hand van een kliksysteem. De houder van de batterij is een bakje waar deze inligt, dit is een iets minder elegantere oplossing omdat we last minute nog van batterijhouder moesten veranderen. Ook hebben we natuurlijk een arm nodig waarop de servo uiteindelijk wordt bevestigd. De arm moest aan een paar voorwaarden voldoen, zo moest er een makkelijke bevestiging zijn voor de servo. Die bevestiging is voorzien op het einde van de arm en is aan de hand van een bakje. Ook moest de afstand van de as van de hoofdmotor en de as van de servo precies 200 mm zijn, dit omdat we met bepaalde verhoudingen zitten om gepaste cycloïden te bekomen. Ten slotte hebben we nog de plaatjes die op de servo moeten komen, deze dienen als tweede roterende beweging om de uiteindelijke cycloïden te bekomen. Ook deze moeten aan precieze afmetingen voldoen, dit opnieuw om die juiste verhoudingen te bekomen (zie ook 8.3).



Figuur 10 : opstelling cycloïden

3.2 3D-printer

Een 3D-printer kan via digitale bouwtekeningen driedimensionale objecten produceren. De printer doet dit door laag na laag op te bouwen. 3D-printen behoort tot de rapid prototyping-industrie. Dit is een techniek waardoor snel fysieke prototypen vervaardigd kunnen worden. Meestal wordt bij een 3D-printer verschillende soorten plastic gebruikt omdat het zeer gemakkelijk plastisch te vormen is. De temperatuur om een "thermoplast" een kunststof die behoort tot de groep van de thermoplasten" plastisch te maken is ook helemaal niet hoog. Dit is ook een groot contrast met andere materialen (een kunststof uit de groep van de thermoharders kan niet gebruikt worden om te 3D-printen). Het is ook zeer goedkoop en is gemakkelijk te verkrijgen. Sommige printers kunnen printen met andere materialen zoals metalen en keramiek.

3.2.1 Geschiedenis

Het driedimensionale printproces is ontstaan uit een aantal projecten van het MIT in samenwerking met het bedrijf 3D Systems in het jaar 1990. Dit was een uitbreiding van rapid prototyping, waarbij het er vooral om ging om snel een prototypemodel te maken. De eerste toepassingen waren met een metaal als productiestof.

Het Massachusetts Institute of Technology heeft na een succesvolle werking licenties overgedragen aan zes bedrijven voor het commercieel produceren en gebruiken van deze technologie. De volgende stap was het produceren van modellen gebaseerd op andere stoffen, zoals keramiek.

In 2005 ontstond er in de Verenigde Staten een snel groeiende markt voor hobbyisten en thuisgebruik met het beschikbaar komen van de open-source RepRap en de Fab@Home projecten. De meeste 3D-printers voor thuisgebruik gebruiken technieken die hiervan afgeleid zijn. Een in 2013 uitgevoerde studie toonde aan dat 3D-printen een massaproduct zou kunnen worden waarmee consumenten geld zouden kunnen besparen bij het kopen van kleine huishoudelijke producten. In plaats van het kopen van een door spuitgieten gemaakt product, zoals een beker of een trechter, zou iemand dat dan thuis kunnen printen.

Er zijn tegenwoordig ook opleidingen die 3D-printen aanbieden in hun lessen, om zo studenten bekend te maken met de mogelijkheden ervan.

3.2.2 Methoden

3.2.2.1 Methode 1: Fijn poeder

Een veelgebruikte methode bij 3D-printen is het gebruikmaken van een fijn poeder (gips, plantaardige stoffen, bioplastic, polyurethaan, polyester, epoxy) waarbij er telkens lagen van dit poeder met elkaar verbonden worden, waarna deze een vaste vorm aannemen. Deze bindingen worden gedefinieerd door een stl.-bestand. Deze methode is de enige methode waarbij een model volledig met kleuren geprint wordt. Het is ook de snelste methode.

Fused deposition modeling (FDM) is een andere manier, waarbij gesmolten polymeren op een supportlaag worden gespoten en het model zo laag na laag wordt opgebouwd.

3.2.2.2 Methode 2: vloeistoffen

Een andere methode is het gebruikmaken van vloeistoffen, zoals fotopolymeer waarbij men ook telkens laag na laag print. Vervolgens wordt er een uv-lamp aan de printkop geactiveerd, die iedere vloeistoflaag vast maakt. Deze methode is ook bekend onder de naam stereolithografie.

Een gelijksoortige printmethode gebruikt een speciale gel die bestraald wordt door een laser op de plekken die vast moeten zijn. Vervolgens wordt het niet-vaste deel van de gel weggewassen. Met deze methode kan men objecten modelleren die kleiner zijn dan 100 nm, alsook complexe modellen.

De tussenlaag die de verschillende lagen bindt, kan men na het maken van het model handmatig verwijderen of mits gebruikmakend van duurdere printertypes, oplossen in een vloeistof, door het model erin te dompelen. Het resultaat is een hol model dat vrij accuraat de bedoelde vorm nabootst.

Het maakt niet uit welke modelleringssoftware men gebruikt voor het model, zolang deze het model kan uitvoeren in het formaat dat gebruikt wordt door de printersoftware.

3.3 Materialen

3.3.1 Filament

Filament is inkt voor een 3D-printer.

Het wordt meestal gemaakt van een thermoplastic. De meest toegepaste plastics zijn polymelkzuur (PLA), ABS en PETG.

3.3.1.1 PLA

PLA is een biobased plastic en een biologische afbreekbare plastic. Biobased betekent dat het vervaardigd is uit natuurlijke producten en niet uit petroleumchemicaliën. Biologisch afbreekbaar betekent dat het terug uiteen kan vallen in melkzuur.

Op school gebruiken we vooral PLA omwille van de volgende eigenschappen:

- goedkoop;
- snel en makkelijk printen;
- redelijk sterk;
- redelijk hittebestendig;
- glanzend;



Figuur 11 : PLA

- beschikbaar in veel verschillende kleuren;
- licht flexibel.

3.3.2 MDF

Medium-density fibreboard of mdf is een plaatmateriaal. In tegenstelling tot bij zacht- en hardboard wordt bij het produceren van een mdf-plaat lijm toegevoegd. De zeer fijngemalen houtvezels worden gemengd met kunstharslijm en in droge toestand geperst. Doordat het geperst is, kan het niet splinteren, dit is dus een voordeel bij het bewerken ervan. Doordat het oplosmiddel van de gebruikte lijm meestal formaldehyde is, zorgt dit ervoor dat mdf jaren na de productie nog altijd formaldehyde uitstoot. Er is geen grote uitstoot, maar als je er continu mee werkt zonder goeie ventilatie kan je beroepsastma ontwikkelen. Bij ons op school worden er mdf-platen gebruikt die een lagere waarde hebben aan formaldehyde. Er zijn ook al testen gedaan in het maaklab van onze school om de verschillende stoffen die vrijkomen bij het laseren op te sporen.



Figuur 12 : MDF

Voordelen

Het grootste voordeel is dat het makkelijk te bewerken is zonder dat je er een duur toestel voor nodig hebt. Tenzij dat je een zeer gedetailleerd stuk wenst. Mdf is ook zeer makkelijk te verkrijgen en kan je in kleine of grote hoeveelheden aankopen. Het heeft een hoog breekpunt, een hoge slijtvastheid, een betere lijmvastheid en een uitzonderlijke dimensionale stabiliteit. De mdf-platen bestaan in veel verschillende groottes en diktes, dus je kan voor gelijk welke toepassing gemakkelijk een geschikte plaat vinden.



Figuur 13 : MDF

Nadelen

Het grootste nadeel aan MDF is dat het niet vochtbestendig is. Er bestaat ook een "groene" mdf die vochtwerend is voor in vochtige ruimtes binnenshuis, maar die kan ook niet buitenhuis gebruikt worden. Nog een nadeel dat we allemaal wel kennen is als je een mdf-plaat gebruikt als plank om dingen op te zetten, zul je zien dat na verloop van tijd deze plank zal doorbuigen. Je zult merken dat als je alles ervan haalt het niet terug recht zal buigen. Dit komt omdat mdf de eigenschap kruip met zich meedraagt. In mdf zit ook wat lijm, deze zorgt deels voor een goeie stevigheid, maar zorgt er ook voor dat beitels en zagen sneller slijten. Als je zoals wij met een lasercutter werkt, heb je inderdaad dat probleem niet. Bij het bewerken van mdf komt er een stof vrij die diep in de longen kan doordringen waardoor men een soort van astma kan krijgen. Nadat het mdf volledig bewerkt is en

klaar om te monteren, is het beter om het eerst te behandelen met lak of verf zodat er geen stof meer kan vrijkomen.

3.3.3 PC-plaat

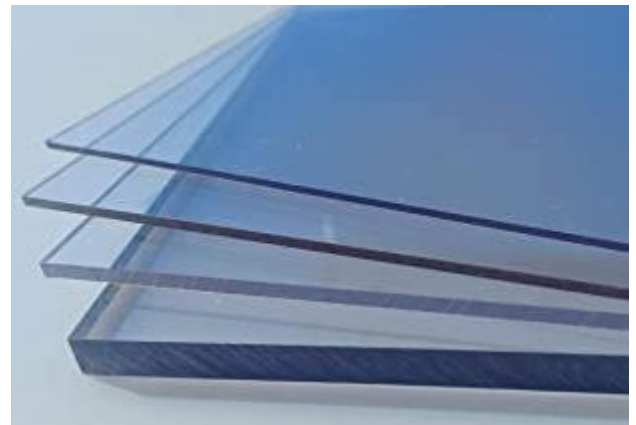
Polycarbonaat is een stevig, hard en helder transparant materiaal dat in tegenstelling tot de meeste andere kunststoffen hoge temperaturen aankan. Het is bijna glashelder en versterkt met glasvezel.

Voordelen

- sterker dan glas
- laat meer licht door dan glas
- is licht in gewicht
- goed bewerkbaar
- duurzaam
- veiliger dan glas

Nadelen

- krasgevoeliger
- niet hittebestendig
- zeer brandbaar
- slagvast, maar splintert bij hoge impact



Figuur 14 : PC-plaat

3.3.4 Easy systems profielen

De easy systems zijn aluminium profielsystemen die flexibiliteit en snelheid garanderen bij het bouwen van constructies. Er zijn ook veel accessoires verkrijgbaar om het monteren eenvoudiger te maken of om extra onderdelen te monteren.

Voordelen

- zeer bekend
- makkelijk verkrijgbaar
- niet te duur
- stevig
- verkrijgbaar in veel verschillende groottes en maten
- snelle en eenvoudige constructie



Figuur 15 : Easy systems profielen

Nadelen

Bij sommige constructies moet je veel uit elkaar halen om een bepaald stuk aan te passen. Er zijn ook bepaalde accessoires waardoor dat makkelijker is, maar die zijn zeer duur. Een ander nadeel is dat bij het zagen er wat metaalschilfers kunnen aanhangen waaraan je je kan snijden.

4. Elektrische studie

In dit onderdeel zullen we spreken over alle elektrische aspecten die komen kijken bij de polyp.

4.1 Pasco-sensoren

4.1.1 Krachtsensor

1) Algemeen

Om de metingen van de centripetale en centrifugale kracht te bekomen maken we gebruik van krachtsensoren. Deze sensoren zijn van het merk PASCO en lezen we binnen in de computer via het programma Capstone. De krachtsensoren bestaan in allerlei vormen en maten. Op school hebben we er twee ter beschikking: de PASCO wireless smart cart en de PASCO wireless force acceleration. Dit zijn beiden sensoren die via bluetooth werken. In onze opstelling gebruiken wij de PASCO wireless smart cart, hiermee kunnen we natuurlijk een kracht meten, maar ook positie, snelheid en versnelling.



Figuur 16 : PASCO wireless smart cart

2) Werking

Om de werking van de krachtsensoren uit te leggen hebben we deze van naderbij bekeken. Dit mag je dan ook tamelijk letterlijk nemen, we hebben de sensoren gedemonteerd. Als we de sensor open hadden liggen, zagen we dat deze een load cel bevatte.



Figuur 17 : Gedemonteerde krachtsensor

Strain gage load cel

Het rekstrookje

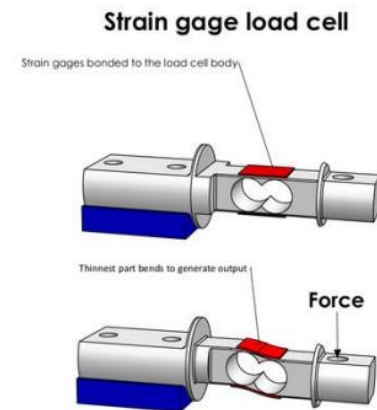
De mechanisch zwakke verbinding

Een load cel is een praktisch voorbeeld van een elektrische brug. Je kunt load cellen opdelen in twee grote groepen: capacitive load cellen en strain gage load cellen. Een capacitive load cel werkt aan de hand van capaciteiten, de vervorming heeft impact op de capaciteit. Bij een Strain gage load cel heeft de vervorming impact op de weerstand van het meetelement. De krachtensor werkt met een Strain gage load cel.

Bij de strain gage load cel wordt op het mechanisch zwakste punt een rekstrookje geplaatst. Dit is een weerstandselement waarbij de waardes veranderen tijdens de vervorming. Dit kunnen we verklaren door de wet van Pouillet:

$$R = \frac{\rho \cdot l}{A}$$

Bijvoorbeeld als het rekstrookje uitrekt, zal de weerstand stijgen. Dit kunnen we verklaren omdat de lengte van het strookje langer zal worden en de oppervlakte van het strookje zal afnemen. Ook omgekeerd zal dit het geval zijn. Het rekstrookje vervormt natuurlijk door de invloed van de kracht.



Figuur 18 : strain gage load cel

4.1.2 Rotatiesensor

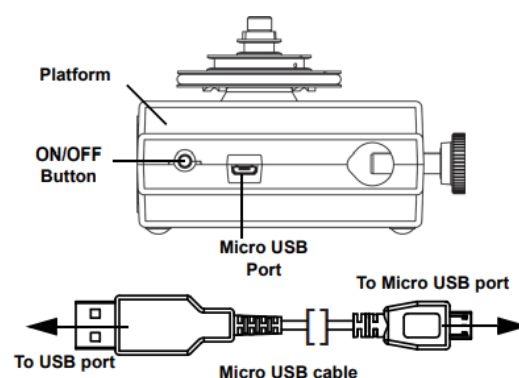
De PASCO Wireless Rotary Motion Sensor is een veelzijdige positie- en bewegingsmetingapparaat. Het meet hoeken met een resolutie van 0,18° en detecteert de bewegingsrichting.

Markeringen aan de buitenkant van de behuizing geven aan wat de standaard positieve richting is. De maximale snelheid ligt tussen de 20 en 80 omwentelingen per seconde.

De sensor meet de hoek, hoeksnelheid en hoekversnelling. Met de as die we op de sensor monteren, kunnen we perfect de hoeksnelheid meten op de hoogte die wij willen.

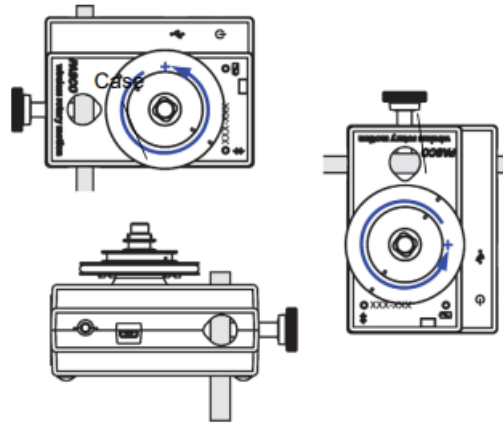


Figuur 19 : rotatiesensor



Figuur 20 : rotatiesensor

De sensorbehuizing heeft twee steunstanggaten die geschikt zijn voor stangen tot 12,7 mm in diameter en de achterkant kan op de steunstang worden geplaatst met behulp van een van de steunstanggaten. Wij gebruiken zo'n gat om te zorgen dat de sensor niet uit de behuizing kan waardoor deze eigenlijk niet kan bewegen. Hierdoor is de sensor perfect georiënteerd voor ons experiment.



Figuur 21 : bevestiging rotatiesensor

4.2 Motor

In de polyp bevinden zich 2 motoren.

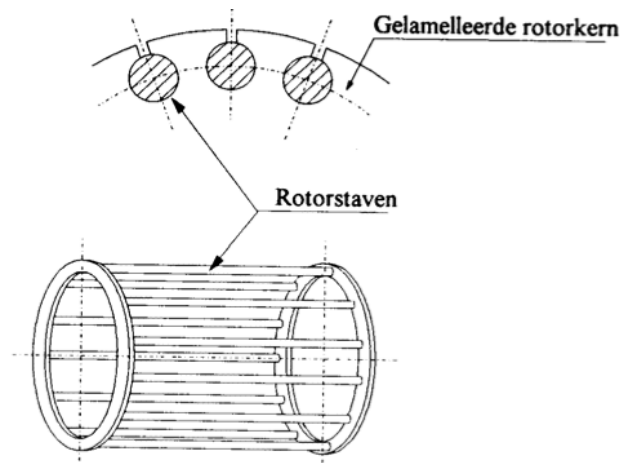
- De hoofdmotor: deze zorgt voor het draaien van de as, en dus de eerste cirkelvormige beweging (in dit deeltje).
- De bij-motor: deze zorgt voor het draaien van de plaatjes, en dus de tweede cirkelvormige beweging (6.5 "servo").

In dit deel zullen we het hebben over de hoofdmotor. Vooraleer we onze hoofdmotor kozen, stelden we enkele voorwaarden op waaraan hij moest voldoen. Zo moest de motor een relatief grote kracht kunnen verzetten bij een laag toerental, met deze criteria gingen we naar meneer Geeraert en zo kwamen we uit bij een driefasige asynchrone motor. Om aan deze eisen te voldoen maakt de motor gebruik van een planetaire versnellingsbak.

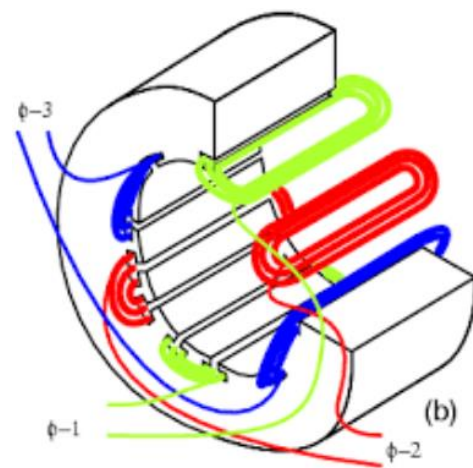
Aan onze motor moeten we een spanning van 24V DC aanleggen, deze spanning wordt dan via een 3-fasige H-brug omgevormd naar wisselstroom (zie 6.2 ESC).

Samenstelling van een driefasige asynchrone motor

Zoals iedere elektromotor heeft een driefasige asynchrone motor ook een rotor en een stator. De stator is het vaste gedeelte van de motor, deze bevat drie spoelen, geschakeld in ster of driehoek. Als we hier een driefasestroom opdringen (bij ons bekomen door een H-brug) zal er een draaiveld gevormd worden. De rotor is het roterende gedeelte van de motor, deze bestaat uit een kooianker. Een kooianker is gemaakt uit verschillende plaatjes staal die op elkaar gemonteerd zijn in een cilindervorm. In die plaatjes zitten er gleuven met koper of aluminium, die aan de uiteinden langs weezijden verbonden zijn.



Figuur 22 : kooirotor

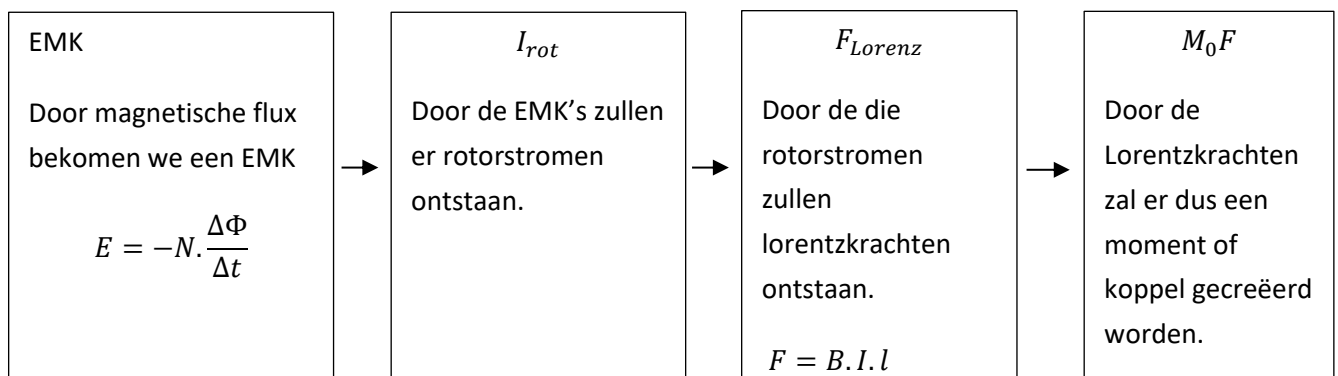


Figuur 23 : driefasewikkeling

Werking van een driefasige asynchrone motor

Zoals al in vorige puntje vermeld, zullen we dus een driefasestroom aan de statorwikkelingen hangen, hierdoor ontstaat een magnetisch veld. Door dit magnetische veld zullen er veldlijnen ontstaan. Het gevolg is dus dat de rotor geleiders door deze veldlijnen zullen gesneden worden. Hierdoor worden EMK geïntroduceerd (zin van de EMK is te bepalen met de rechterhandregel). Omdat de rotor geleiders een gesloten kring vormen, zullen hierin inductiestromen vloeien. Er ontstaan dus stroomvoerende geleiders in een magnetisch veld, hierdoor zullen er Lorentzkrachten kunnen ontstaan (zin van de Lorentzkrachten is te bepalen met de linkerhandregel). De resulterende Lorentzkracht zal het koppel van onze motor vormen, en zal dan ook onze rotor laten roteren.

Werking samengevat



De werking bij een belasting

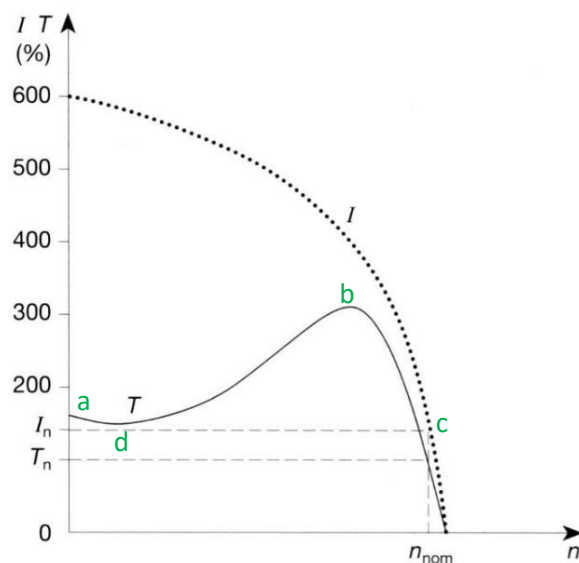
De gebruiker van de motor zal bij een belasting een groter motorkoppel vragen. De rotor zal hierdoor dus tegengehouden worden door deze belasting, het rechtstreekse gevolg is dus dat de rotor meer veldlijnen zal snijden. Hierdoor zal de rotor EMK vergroten, waardoor ook de inductiestroom en dus ook de Lorentzkrachten zullen toenemen. Dit zal dan resulteren in het toenemen van het motorkoppel.

Slip en rotatiefrequentie van het draaiveld

We weten dat de rotor minder snel zal ronddraaien dan de fluxvector, de rotor slipt als het ware ten opzichte van het stator draaiveld: $n_r < n_s$

Slip kunnen we dus schrijven als: $s = \frac{n_s - n_r}{n_s}$

Koppel en rotatiefrequentie karakteristiek



Figuur 24 : koppel-rotatiekarakteristiek

Beschrijving

- Motor start: slip = maximaal en aanzetkoppel = klein.
- Kipkoppel = maximaal koppel.
- Nominaal rotatiefrequentie n_s met daarbij horend nominaal koppel T_n . Dit is het toerental dat de motor 24/24 7/7 kan draaien.
- Het minimum koppel ook wel zadel genoemd.

Het gebied van a tot b is een onstabiel gebied, een kleine verandering in de belasting kan voor een relatief grote snelheidsverandering zorgen. Het gebied boven c is het koppel en toerental dat de motor kortstondig kan geven, dit omdat de windingen door de grote stromen te veel opgewarmd zouden worden bij lang gebruik.

Aanloop van de driefasemotor

We zien ook in bovenstaande grafiek dat de stroom in functie van het toerental is weergegeven. In de startfase zien we dat we een aanzetkoppel hebben, maar dat de aanloopstroom verschillende keren groter is dan de nominale stroom I_n . Hoge startstromen belasten het net sterk en veroorzaken spanningsdalen, om dit tot een minimum te beperken zullen we de aanloopstroom limiteren. Dit doen

we via een aanloopmethode, bij ons is dit via een frequentiesturing. Het onderdeel dat dit voorziet, is de ESC (zie ook 6.2), hier zie je een light versie.

Een frequentiesturing bestaat uit drie grote blokken:

1^e blok

De gelijkrichter, gebeurt via een brugschakeling.

2^{de} blok

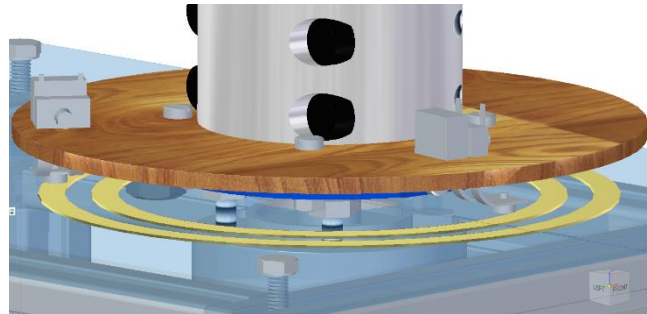
De afvlakking van de stroom, gebeurt via een condensator.

3^{de} blok

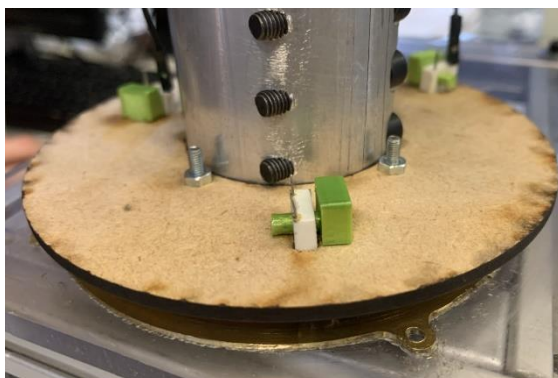
Van gelijkstroom terug naar wisselstroom gaan, ook wel INVERTOR genoemd.

4.3 Sleepcontacten

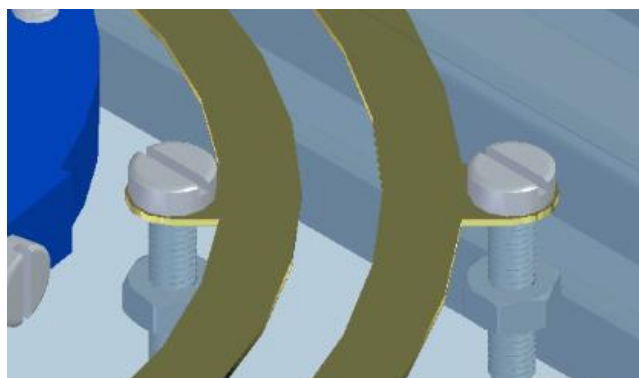
Omdat we in de polyp met 2 cirkelvormige bewegingen zitten, hebben we een systeem nodig dat elektriciteit van het vaste gedeelte naar het ronddraaiende gedeelte brengt, dit om de Arduino en dus ook de servomotor van elektriciteit te voorzien. Hiervoor pasten we de sleepcontacten toe. Deze sleepcontacten bestaan uit twee messingringen op het vaste gedeelte en drie sleepcontactjes op het ronddraaiende gedeelte. Aan een van de twee messingringen zullen we de positieve pool van de bron hangen, aan de andere de negatieve pool. Als de polyp zal beginnen roteren, zullen de drie sleepcontactjes over de twee messingringen slepen, zo zal de elektriciteit overgebracht worden. We hebben voor drie sleepcontactjes geopteerd omdat we er vanuitgaan dat drie punten een vlak beschrijven en dat er zo altijd contact zal zijn tussen de sleepcontactjes en de sleepringen. Op de messingringen hebben we twee uitsparingen voorzien, dit om de positieve en negatieve pool van de bron op aan te sluiten. Deze zullen we monteren aan de hand van twee boutjes.



Figuur 25 : zijaanzicht sleepringen



Figuur 26 : constructie sleepringen

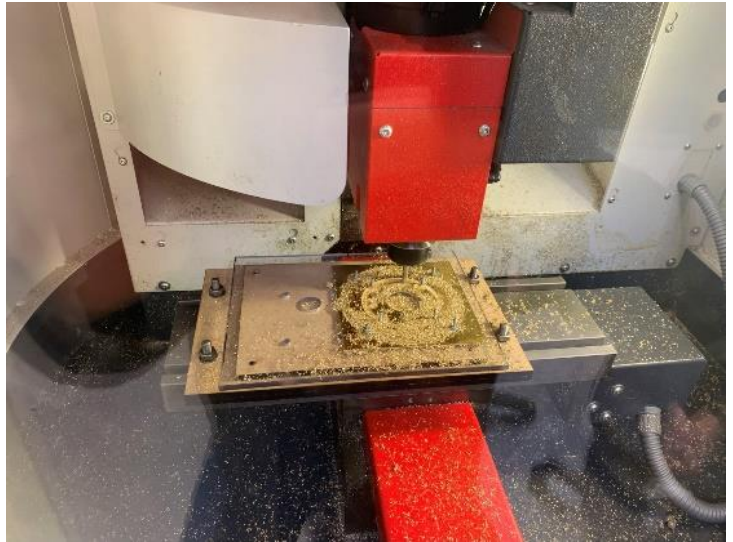


Figuur 27 : aansluiting elektriciteit

4.3.1 Constructie van de sleepcontacten

1. De messingringen

Het oorspronkelijke idee van de messingringen was om deze te etsen, dit was een snellere en makkelijkere oplossing dan ze te frezen. Alleen zitten we bij het etsen met het probleem dat de koperdikte maar $35\mu\text{m}$ is. De tip kwam dan ook van de jury om dit niet te hanteren, omdat door de constante wrijving van de sleepcontacten en het koper, dit koper zou kunnen wegslijten. De messingringen hebben we gefreesd samen met de plexiplaat die op de profielen ligt. Dit hebben we als volgt gedaan, eerst kleefden we een vierkant stuk messing van $130\times 130\times 1$ met dubbelzijdige tape op de plexiplaat. Daarna freesden we alle gaten uit de plexiplaat en de messingringen. Verder maakten we een mal waarin we de plexiplaat konden vastbouten, hierdoor zat de plaat dus vast en konden we de twee ringen frezen. Ten slotte hebben we de contouren van de plexiplaat gefreesd.



Figuur 28 : frezen van sleepringen

2. De sleepcontactjes

De sleepcontactjes hebben we gehaald uit een oude relais, deze zullen op de messingringen slepen. Aan de sleepcontactjes hebben we ook nog een stukje litzeband gesoldeerd, dit om optimaal raakcontact te bekomen. De contactjes zelf zitten vast aan een MDF-plaatje via een ge-3D-print stukje. Dit plaatje zit dan weer aan een ge-3D-print bevestigingsstuk en dit bevestigingsstuk zit op zijn beurt dan weer vast aan het klemstuk.



Figuur 29 : bevestigen sleepringen

4.4 Batterij

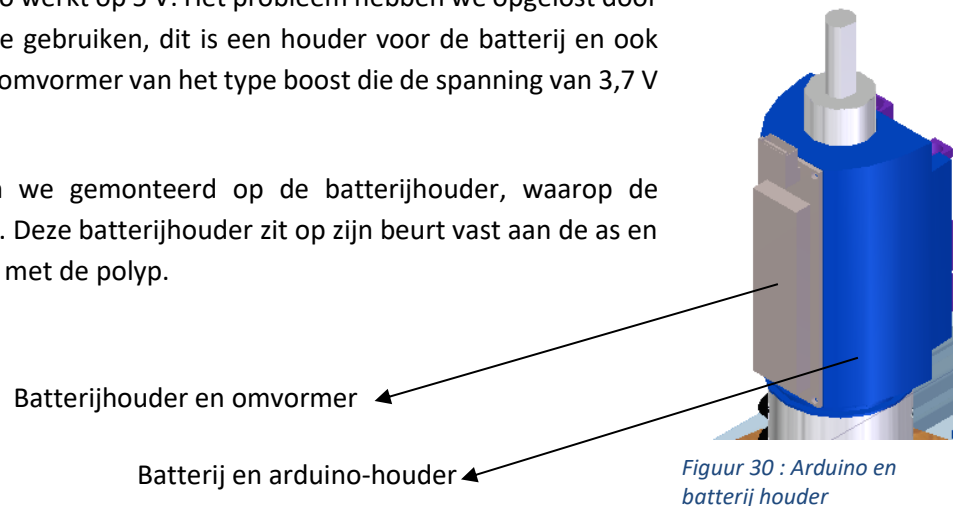
4.4.1 Praktijk/gebruik

De polyp maakt gebruik van 2 Arduino Uno's, eentje op het vaste gedeelte en eentje op het draaiende gedeelte. De Arduino Uno op het draaiende gedeelte zullen we voorzien van spanning via de sleepcontacten (zie puntje 5.2).

Om een constante spanning te bekomen hebben wij ervoor gekozen om een back-up batterij te gebruiken. Zo kan de Arduino Uno nooit zonder spanning vallen.

De batterij waarvoor wij opteren is eentje van het type: ICR18650. Dit nummer duidt op het formaat, deze batterij heeft een diameter van 18.6mm en een lengte van 65.2mm. De 0 op het einde geeft aan dat de batterij cilindervormig is. Het is een herlaadbare batterij die op 3,7 V werkt. Dit is een probleem want de Arduino Uno werkt op 5 V. Het probleem hebben we opgelost door een Battery shield te gebruiken, dit is een houder voor de batterij en ook meteen een DC/DC-omvormer van het type boost die de spanning van 3,7 V naar 5 V omzet.

De batterij hebben we gemonteerd op de batterijhouder, waarop de Arduino Uno ook zit. Deze batterijhouder zit op zijn beurt vast aan de as en zal dus mee roteren met de polyp.



4.4.1.1 Battery shield

De battery shield dient als houder voor de batterij, maar bevat ook de elektronica om de batterij op te laden en om van 3,7 V naar 5 V te gaan via een boost converter.

Een 18650 Battery Shield V3 is een apparaat dat is ontworpen om een 18650 oplaadbare lithium-ion batterij te gebruiken als voedingsbron voor andere elektronische apparaten. Het wordt vaak gebruikt in elektronica-projecten waarbij een draagbare en oplaadbare voedingsoplossing nodig is.



Figuur 31 : battery shield

Hier zijn enkele kenmerken en functionaliteiten van een typische 18650 Battery Shield V3:

- bevat batterijhouder
- oplaadfunctie

Het shield heeft een ingebouwde oplaadmodule die de 18650 batterij kan opladen wanneer deze is aangesloten op een voedingsbron, zoals een USB-poort. Dit maakt het gemakkelijk om de batterij op te laden zonder deze uit de shield te hoeven verwijderen.

- spanningsregeling

De 18650 Battery Shield V3 heeft vaak een ingebouwde spanningsregelaar die de uitgangsspanning aanpast naar een vast niveau, bijvoorbeeld 5 Volt. Dit maakt het mogelijk om andere apparaten met een constante spanning van de batterij te voorzien.

- USB-uitgang

Het shield kan vaak worden uitgerust met een USB-poort, waardoor het gemakkelijk is om andere apparaten op te laden of van stroom te voorzien met behulp van de opgeslagen energie in de batterij.

- Bescherming

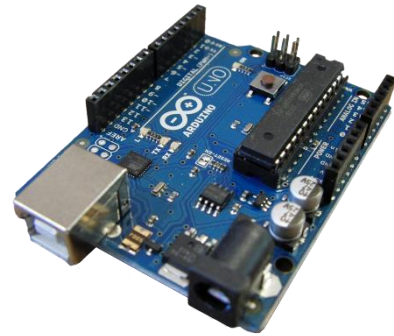
Een 18650 Battery Shield V3 biedt meestal ingebouwde beveiligingsfuncties, zoals kortsluitbeveiliging, overbelastingsbeveiliging en over-ontladingsbeveiliging, om de batterij en aangesloten apparaten te beschermen tegen mogelijke schade.

Het exacte ontwerp en de specificaties van een 18650 Battery Shield V3 kunnen verschillen afhankelijk van de fabrikant of het specifieke model. Het is altijd belangrijk om de documentatie van de fabrikant te raadplegen voor specifieke details en instructies met betrekking tot het gebruik van de shield.

5. Elektronische studie

5.1 Arduino

We hebben ook een makkelijke manier nodig om alle componenten van de polyp aan te sturen. Hiervoor gebruiken we een arduino, specifiek een arduino UNO. Een arduino is eigenlijk een soort van 'hardware' die ervoor zorgt dat je alles via een gemakkelijke programmeeromgeving kan aansturen en inlezen. Merk op dat we hier hardware tussen aanhalingstekens zetten, dit is omdat een arduino eigenlijk geen hardware is, maar een plaatje dat verschillende microcontrollers en componenten verbindt. Bij bijvoorbeeld de arduino UNO is de microcontroller de 'ATMega328', deze zorgt eigenlijk voor al het aansturen bv. welke pin wanneer wordt hoog of laag gemaakt. De microcontroller zal doen wat jij ernaar doorschrijft via de programmeeromgeving van arduino die gebruik maakt van de programmeertaal 'C++'. De programmeeromgeving die je gebruikt, heet Arduino IDE en we programmeren hiermee de volledige polyp. Op de foto zie je ook dat er 2 manieren zijn om de arduino te voeden namelijk via de USB B-connector waardoor je ook de code doorschrijft via de computer of via de power port die je dan met een speciale adapter via het stopcontact kan voeden. Je kan ook zien dat er digitale pinnen en analoge pinnen zijn. Digitale pinnen zijn om een 0 of 1 of dus een laag of hoog signaal te ontvangen/versturen. Deze pinnen worden gebruikt om bijvoorbeeld drukknoppen in te lezen of leds aan te sturen. Sommige van deze pinnen ondersteunen ook het versturen of ontvangen van PWM-signalen om bijvoorbeeld een servomotor aan te sturen. De analoge pinnen hebben geen hoog of laag, maar kunnen veel waarden doorkrijgen zoals 0 tot 1023 van een potentiometer. De 0 tot 1023 zijn gedigitaliseerde waarden, in werkelijkheid staat er over de pinnen van de arduino 0 V tot 5 V en kan de arduino zo een waarde hieruit halen. De pinnen die niet aangeduid zijn, zijn onder andere de 5 V, 3.3 V en GND-pinnen, deze gebruik je om de componenten die je aanstuurt te voeden. Voor sommige componenten zoals grote motoren kan de arduino nooit genoeg voeden en heb je dus een externe voeding nodig.



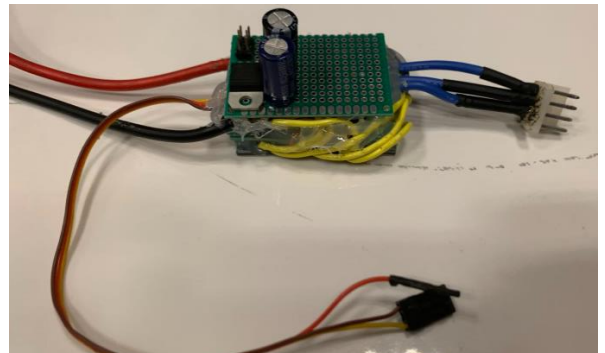
Figuur 32 : Arduino



Figuur 33 : Arduino-poorten

5.2 ESC

Om de snelheid van de hoofdmotor te regelen maken we gebruik van een ESC of 'Electronic Speed Controller'. Deze zal ervoor zorgen dat we via elektronica de frequentie van de stroom kunnen aanpassen en zo dus de snelheid van de motor kunnen regelen (zie ook 5.2). Een ESC staat over het algemeen ook bekend als een frequentieregelaar of een VFD (Variable Frequency Drive), hiermee kan je dus door signalen door te sturen naar de ESC de frequentie van de stroom die uit de ESC komt aanpassen zodat de

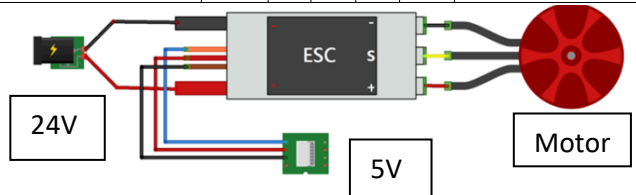


Figuur 34 : ESC

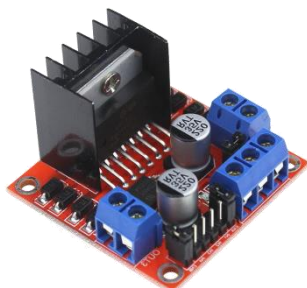
snelheid van de motor geregeld wordt. Een motor zal een grotere hoeveelheid stroom trekken naarmate deze meer belast wordt, deze stroom moet ook door de FET's (Field Effect Transistors) in de ESC vloeien waardoor de ESC

Drain-to-Source Breakdown Voltage	BV_{DSS}	60	—	—	V	$V_{GS} = 0V, I_D = 10 \mu A$
Gate Threshold Voltage	$V_{GS(th)}$	0.8	—	3	V	$V_{GS} = V_{DS}, I_D = 1 \text{ mA}$
Gate Body Leakage Current	I_{GSS}	—	—	10	nA	$V_{GS} = \pm 15V, V_{DS} = 0V$
Zero-Gate Voltage Drain Current	I_{DSS}	—	—	1	μA	$V_{GS} = 0V, V_{DS} = 48V$
		—	—	1	mA	$V_{GS} = 0V, V_{DS} = 48V, T_A = 125^\circ C$ (Note 1)
On-State Drain Current	$I_{D(ON)}$	75	—	—	mA	$V_{GS} = 4.5V, V_{DS} = 10V$
Static Drain-to-Source On-State Resistance	$R_{DS(ON)}$	—	—	5.3	Ω	$V_{GS} = 4.5V, I_D = 75 \text{ mA}$
		—	—	5	Ω	$V_{GS} = 10V, I_D = 500 \text{ mA}$

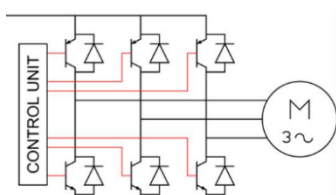
vrij warm kan worden. Een transistor is een elektronische stroomversterker, maar in onze toepassing gebruiken we deze als schakelaar. Zoals je hiernaast op de datasheet kan zien, is er een $R_{DS(ON)}$ waar stroom door zal vloeien. Als gevolg van deze stroom zal er ook spanning over de weerstand staan en zal er dus een vermogen zijn waardoor de ESC zal opwarmen. De grootte van dit vermogen zal dus afhangen van hoe groot deze $R_{DS(ON)}$ is. Om de componenten in de ESC te beschermen tegen de warmte is onderaan de ESC een ijzeren plaatje, dat als koelplaatje dient, bevestigd. Dit zal ervoor zorgen dat de ESC voldoende warmte kan afgeven en de componenten dus niet zullen stukgaan.



Figuur 35 : ESC schematisch



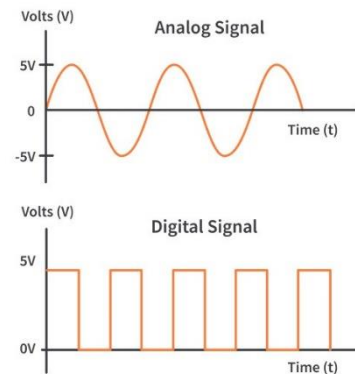
Figuur 36 : H-brug



Figuur 37 : H-brug schematisch

De ESC wordt gevoed door gelijkstroom of DC, maar de motor werkt eigenlijk op 3-fasige wisselstroom (vandaar 3 draden). De driefasige wisselstroom kunnen we maken door te werken met een 3-fasige H-brug. Hiermee kunnen we eigenlijk ook direct de frequentie van de stroom regelen. Deze H-brug zit ingebouwd in de ESC en creëert 3-fasige wisselstroom door middel van elektronische schakelaars of FET's, deze kunnen heel snel en betrouwbaar schakelen waardoor je de frequentie van de stroom kan regelen. In de ESC zit ook elektronica die signalen die je via een microcontroller bv. Arduino UNO verstuurt omzet naar de juiste frequentie om de motor op een bepaalde snelheid te laten draaien. De naam van deze signalen zijn PWM (Pulse-Width Modulation)-signalen.

In de ESC zit elektronica (FET's) die op signaal van de Arduino zullen schakelen om een 3-fasige stroom te creëren. Dit gebeurt door middel van PWM-signalen die de Arduino doorstuurt naar de ESC. De PWM-waarden hebben een vaste periode, maar je kan de lengte van de hoge signalen variëren. Als de PWM-waarde die je doorstuurt naar de ESC verandert, zal de frequentie van de 3-fasige



Figuur 38 : PWM-signalen

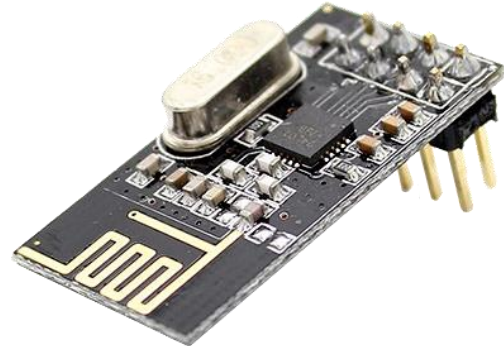
stroom die uit de ESC zal komen, ook veranderen. Dit kunnen we dan gebruiken als aanzetmethode. De snelheid van de motor wordt bepaald door de slip van de motor en hoeveel de motor belast is. Deze 2 dingen zullen een motor vertragen, je hebt dan ook nog de versnelling van de motor wat bepaald wordt door de grootte en de frequentie van de stroom.

Wat is een aanzetmethode?

Een aanzetmethode wordt eigenlijk toegepast om de aanzetstroom, die 7 à 9 keer groter is dan de nominale stroom, te verkleinen. Je hebt hiervoor verschillende methodes zoals een PLC of een softstarter. De methode die wij gebruiken, is een frequentiesturing of dus een ESC. Bij het vertrekken van de motor kan je dus met de ESC het draaiveld vertragen zodat de koppel-toerentalgrafiek samengedrukt wordt. Omdat je de motor vertraagt, zal je motor zal veel minder lang doen om op snelheid te komen waardoor deze aanzetstroom veel minder lang aanhoudt en het aanzetten van de motor dus zuiniger is. Dan kan je met de ESC de motor opnieuw versnellen wat gepaard gaat met een stroomverhoging. Als je dit geleidelijk aandoet, beperk je de stroomafname wat dit een aanzetmethode maakt. Met de ESC komt natuurlijk ook nog de bonus dat je de snelheid van de motor kan regelen.

5.3 NRF 24

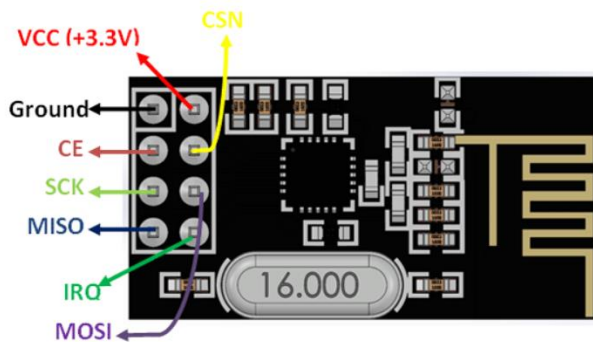
Op de polyp hebben we een draaiend deel en een stilstaand deel, hiermee gaan natuurlijk ook een paar problemen gepaard. Een eerste probleem is om de Arduino en de servomotor op het draaiende deel te voeden, dit hebben we opgelost met de sleepringen. Het tweede probleem is een gevolg van het eerste, bij de sleepringen is er soms slecht contact waardoor het overbrengen van data zeer moeizaam zal verlopen. Om deze data door te sturen zullen we dus draadloos werken, wat veel beter zal lukken. Dit kunnen we op verschillende manieren doen zoals met bluetooth, via wifi ... Hiervoor gebruiken we de NRF24, deze doet



Figuur 39 : NRF 24

perfect wat nodig is namelijk data doorsturen. Dit is echter wel beperkt tot een aantal bytes, maar meer hebben wij niet nodig. Een NRF24 is eigenlijk een module die we kunnen verbinden met onze Arduino aan zowel de receiver- als de transmitterkant zodat deze via radiogolven kunnen

communiceren met elkaar. In het programma van de transmitter zit een soort van pakketje data die we dan via de NRF24 bij de transmitter naar die van de receiver doorsturen. Aan de kant van de receiver kan je in de code eigenlijk data uit dit pakketje halen en gebruiken om in ons voorbeeld de servomotor aan/uit te zetten of sneller/trager te laten draaien. NRF staat voor 'Nordic Radio Frequency' en de 24 in de naam staat voor 2,4 GHz, dit is de frequentie waarop de NRF24 signalen zal versturen. Op de foto hiernaast kan je zien dat de NRF24 8 aansluitpinnen heeft. De bovenste 2 dienen voor voeding, hij wordt



Figuur 40 : NRF 24 schematisch

gevoed op 3,3 V en er is natuurlijk ook een massa.

Daarnaast heb je 3 aansluitingen die met de host controller zullen communiceren:

De communicatie met de host controller gebeurt via een SPI-bus (**S**erial **P**eripheral **I**nterface). Dit is een seriële bus tussen ten minste 2 mediums, typisch een 'master' en een of meerdere 'slaves'.

-MOSI (**M**aster **O**ut **S**lave **I**n): deze pin zorgt voor communicatie van de host controller naar de NRF24

-MISO (**M**aster **I**n **S**lave **O**ut): deze pin zorgt voor communicatie van de NRF24 naar de host controller

-SCK (**S**erial **C**lock): dit is de transmissieklok van host controller naar NRF24.e

Dan zijn er nog 3 pinnen waarvan de eerste 2 ook essentieel zijn voor een goede werking.

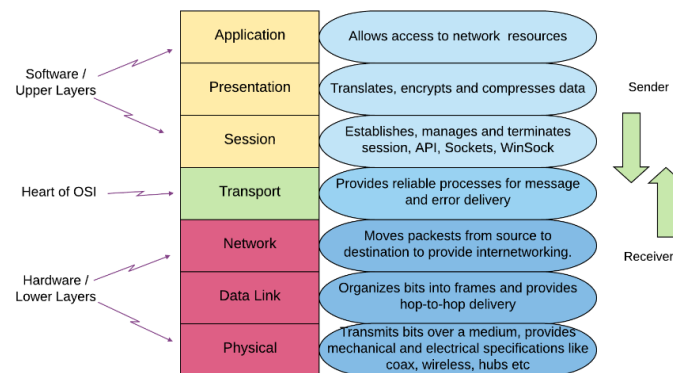
-CSN (**Chip Select iNverted**): soms kunnen er op de SPI-bus meerdere slaves aanwezig zijn zoals eerder vermeld. Het laag maken van deze pin zorgt ervoor dat de NRF24 kan deelnemen aan buscommunicatie.

-CE (**Chip Enable**): Deze pin moet hoog zijn zodat de NRF24 zou kunnen werken. Zo niet, dan zal de NRF24 in ultra low power (900nA)- mode gaan.

-IRQ (**Interrupt ReQuest**): De NRF24 kan deze pin hoog/laag maken om bij ontvangst van data de host controller te informeren van ontvangst. Deze pin is niet verplicht aan te sluiten, maar wij gebruiken deze wel.

Het is ook nog belangrijk om te melden dat de SCK, MOSI en MISO respectievelijk op pinnen 13, 11 en 12 verbonden moeten worden aan de Arduino UNO. Je moet ook de ground aan de ground van de Arduino verbinden en zeker aan 3,3 V verbinden, niet aan 5 V. Voor de pinnen CE en CSN mag je perfect kiezen op welke pinnen je ze aansluit, dit maakt geen verschil als je ze juist definieert in het programma. Dan is er ook nog de IRQ-pin die specifiek voor de Arduino UNO op pin 2 of 3 aangesloten mag worden. Dit is wel anders voor elke microcontroller.

Zoals eerder vermeld zal de NRF24 gebruik maken van radiogolven om de data door te sturen van transmitter naar receiver. De NRF24 zal hiervoor werken op de frequentie van de ISM-band namelijk 2,4GHz. De ISM-band wordt vooral gebruikt voor industriële, wetenschappelijke en medische doeleinden, ISM staat dan ook voor 'Industrial, Scientific and Medical'. Achter het versturen van data zit ook een protocol net zoals achter al het versturen van data in de wereld zoals 4G en wifi. Om dit te beschrijven gebruiken we het OSI-model: dit model is een hulpmiddel om complexe zaken op te delen in gemakkelijk begrijpbare stukken, het helpt de dingen die gebeuren op te delen in 'verteerbare' brokken, dit model bestaat uit 7 lagen waarvan het laagste het verzenden van pure logische 0-en en 1-en (dus via datakabels of draadloos zoals wij het doen). De NRF24 zal gebruik maken van een protocol dat zich situeert op 3 lagen: 1, 2 en 7. Dit protocol noemt het 'shockburst'-protocol en is opgedeeld in 5 delen zoals je kan zien op de foto hiernaast. Het deeltje dat in laag 7 zit, staat open voor input van de gebruiker. Hierin zullen de data zitten die de gebruiker wil doorsturen van transmitter naar receiver.



Figuur 41 : OSI model

Veld	Grootte	OSI-laag	Nut
Preamble	1 byte	1 (datalink)	Aanduiding begin van nieuw bericht
Address	3-5 bytes	2 (netwerk)	Addressering van ontvanger
Packet control	9 bytes	1 (datalink)	Beschrijving frame opbouw
Payload	0-32 bytes	7 (applicatie)	Vrij in te vullen veld voor de gebruiker
CRC	1-2 bytes	1 (datalink)	Foutcontrole corrupte data

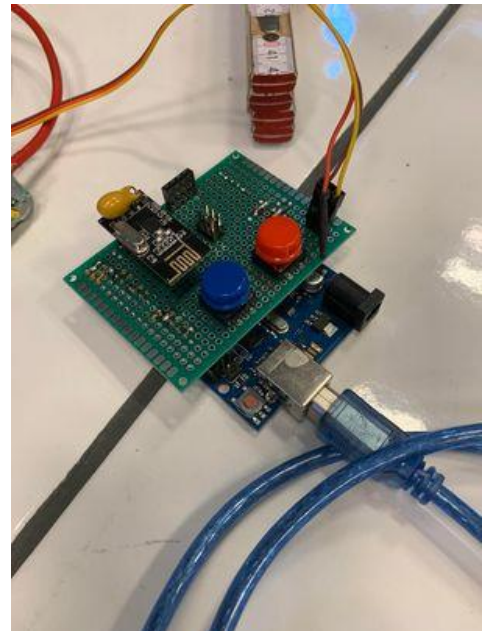
Figuur 42 : protocol van NRF 24

De makers van deze module hebben ook nog gekozen om een tweede protocol aan de NRF24 toe te voegen. Dit protocol heet het 'packet control' veld, hierin zitten ook 2 delen van laag 4. De reden dat ze dit protocol nog hebben toegevoegd is om te controleren of de communicatie tussen de 2 NRF24's goed verloopt en

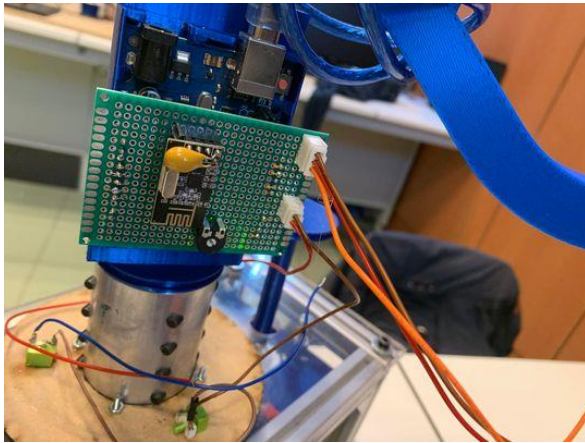
kan dus ook een foutmelding terugsturen als de communicatie niet goed gaat. Als dit protocol niet in de controller zou zitten, zou je via de software deze controle erin moeten programmeren. Dit protocol zit vooral in de controller om de software wat simpeler te maken.

5.4 Shields

Voor alle elektrische verbindingen tussen bijvoorbeeld de pinnen en de motoren of het aansluiten van de knoppen aan de Arduino hebben we jumperdraadjes nodig. Er moeten ook dingen zoals een NRF24 aangesloten worden waarvoor je 8 jumperdraadjes nodig zou hebben. Dit wordt op den duur zeer ingewikkeld en de draadjes komen ook vrij gemakkelijk los. Om dit op te lossen maken we iets wat een 'shield' heet, een shield zal zorgen dat er elektrische verbindingen zijn tussen alle componenten die verbonden moeten worden met de Arduino. Hiervoor gebruiken we een groen printplaatje met pinnen aan de onderkant gesoldeerd om op de Arduino te klikken. Dan hebben we met kleine draadjes de componenten in verbinding met de spanning en de juiste pinnen gezet. Dit zorgt ervoor dat alles dus veel handiger is en als bijgevolg er ordelijker uit ziet. Er zijn ook verschillende pinnen op het bordje om in verbinding te brengen met de motor, de NRF's ... We hebben een shield gebouwd voor zowel de transmitter als de receiver. Bij de transmitter staan 2 knoppen op het bord die dienen als aan/uit-knop, 2 pinnen om de grote motor aan te verbinden wat de massa is en de pin waarnaar hij zal doorsturen, er staan ook nog 6 pinnen om de connector van de grote motor op te verbinden. Hiermee lezen we de puls in die de motor uitstuurt waarmee we dan ook de snelheid van de servomotor kunnen aanpassen. Er staan dan ook nog 4 pinnetjes bij die dienen als 'uitbreiding', je kan er een bluetooth-module op aansluiten en zo de polyp via bluetooth aansturen. Er zijn ook 8 pinnen voor de NRF voorzien.



Figuur 43 : transmitter



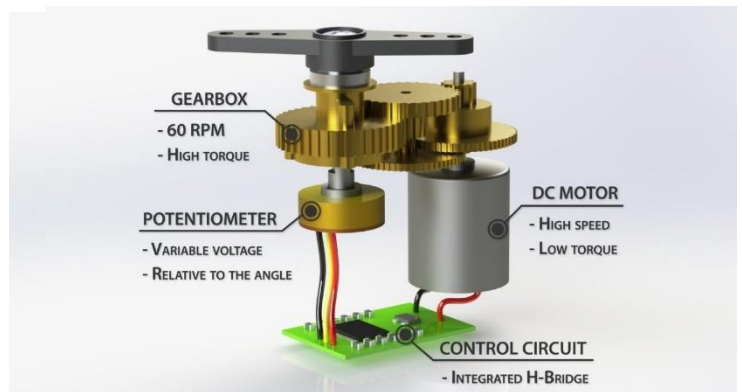
Figuur 44 : receiver

Dan is er ook nog de receiver, deze bevat als eerste 8 pinnen voor de NRF om te communiceren met de transmitter. Hij bevat ook een connector van 3 pinnen om de servomotor op de arm aan te sturen en een connector van 2 pinnen om de Arduino te voeden via de sleepcontacten. We voeden de Arduino ook via de batterij die dus een soort van 'buffer' is. Als laatste staat er ook nog een potentiometer op waarmee we de PWM-waarde van de servomotor in stilstand kunnen veranderen aangezien deze waarde met de temperatuur verandert.

5.5 "Servo"

5.5.1 Algemeen

Om de 'karretjes' te laten draaien maken we gebruik van een servomotor op het einde van de arm. De reden dat we dit soort motor gebruiken is omdat deze gemakkelijk aan te sturen is met een arduino UNO, relatief licht is en dus ideaal is om op het einde van de arm te plaatsen. Een servomotor kan je eigenlijk continu en niet continu gebruiken. Als je een servomotor niet continu gebruikt, zal hij een referentiesignaal opwekken. Daarna stuur je een bepaald signaal door, verschilt dit signaal van het referentiesignaal dan zal hij naar links of rechts draaien. Zijn de signalen gelijk, dan stopt de motor. Niet continu servomotoren kunnen maar 180° naar elke kant draaien, zo kan je dus door de juiste signalen door te sturen ook de hoek instellen die je wenst. De motor die dit alles aan het draaien maakt, is een DC-motor die, zoals de naam het zegt, zal gevoed worden op gelijkspanning. Als we hier zonder referentiesignaal en potentiometer werken, kunnen we deze blijven doordraaien en is de servomotor continu. Wij gebruiken de motor op deze manier en kunnen hem dus aansturen door met PWM-(uitgelegd bij de ESC) signalen te werken en zo de snelheid van onze servomotor aan te passen. De DC-motor die in de servomotor zit, is een motor met borstels aangezien onze servomotor spotgoedkoop van design is. Dit is dan ook de reden dat de motor zeer onvoorspelbaar is in snelheid bij een bepaalde PWM waarde en ook vrij snel zal stukgaan.



Figuur 45 : servo

5.5.2 Hoeksnelheid van de servomotor

Om de cycloïden te bekomen die we voor ogen hebben is het natuurlijk van groot belang om van beide motoren de hoeksnelheid te weten. Dit hebben we dan ook moeten bepalen van de servomotor.

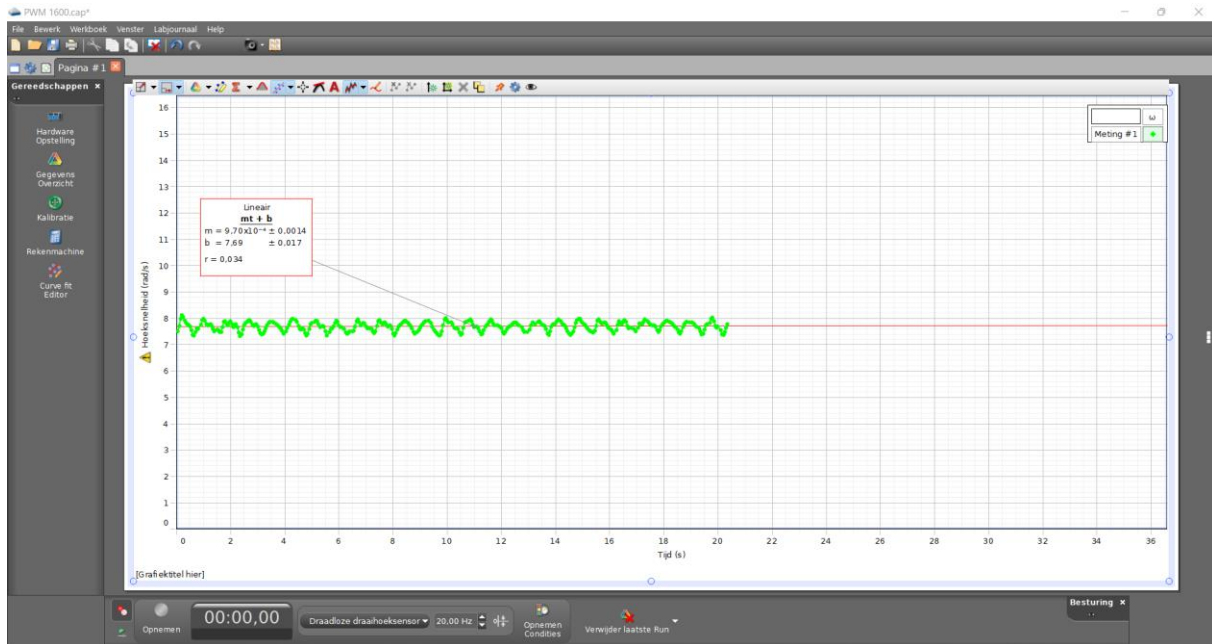
Hoe zijn we te werk gegaan?

Zoals hierboven uitgelegd kunnen we een servo aansturen door er verschillende PWM-signalen naartoe te sturen. Elk PWM-signaal staat voor een verschillende snelheid. Zo hebben we dus bij verschillende PWM-signalen (gelegen tussen 1550 en 1950) de hoeksnelheid gemeten. Hiervoor maakten we eerst een proefopstelling waarbij we de rotatiesensor en de servomotor aan een statief bevestigden. Eenmaal we een geschikte opstelling hadden, zijn we begonnen met meten. De hoeksnelheid hebben we gemeten met de rotatiesensor van PASCO, al de info lazen we binnen op de computer via het programma Capstone.



Figuur 46 : Meetopstelling

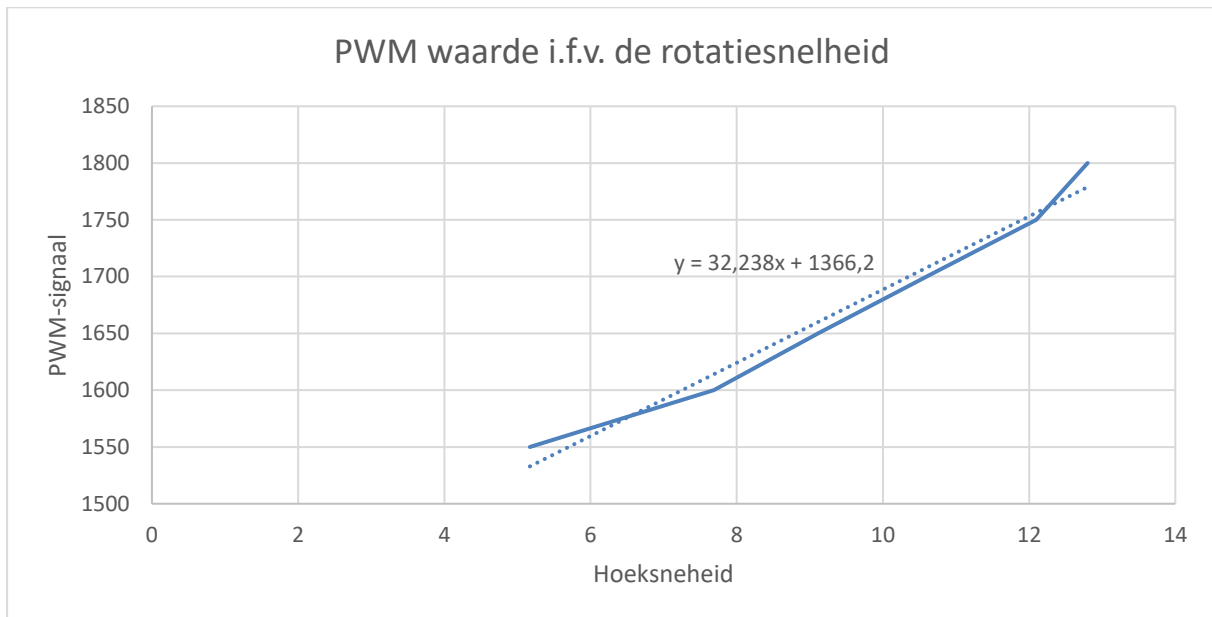
Op deze foto zien we onze opstelling, deze is geconstrueerd door middel van twee statieven. Aan eentje hangt de roterende servo, aan de ander hangt de rotatiesensor. Beiden zijn verbonden door middel van een elastiek.

Meetresultaten:

Figuur 47 : hoeksnelheid i.f.v. tijd

Hier hebben we een voorbeeld van zo'n meting. Zoals je ziet, leest het programma alle informatie binnen in een rekenblad. Op de X-as hebben we de tijd geplaatst in seconden en op de Y-as plaatsten we de hoeksnelheid in radialen per seconde. Bij elke meting hebben we ongeveer een tijdsinterval van twintig seconden genomen. Eenmaal de meting gedaan was, konden we via het programma hieraan een lineaire functie koppelen van de vorm: $y(t) = mt + b$. We merkten op dat dit altijd een bijna constante functie was (door meetfouten was deze niet exact constant). Zo konden we gemakkelijk de hoeksnelheid aflezen op de Y-as.

Deze meting hebben we in totaal acht keer uitgevoerd. De PWM-waarden varieerden van duizendvijfhonderd tot duizendnegenhonderdvijftig en versprongen iedere meting met vijftig.

Verwerking van gegevens:

Figuur 48 : PWM i.f.v. rotatiesnelheid

Deze meting hebben we in totaal vijf keer uitgevoerd. Als we de meetresultaten hadden, hebben we deze uitgewerkt in een grafiek met de PWM-waarde i.f.v. de rotatiesnelheid. Dit is omdat we bij het maken van onze cycloïden altijd de gewilde rotatiesnelheid van de servo weten en we moeten hier dus het overeenstemmende PWM-sigitaal zien van te vinden.

We zien dat er een lineair verband is tussen de PWM-waarden en de rotatiesnelheden. Het bekomen functievoorschrift is van de vorm: $y(x) = Ax + B$ ofwel $y(x) = 29,374x + 1387,6$. We willen cycloïden bepalen met onze polyp, wat betekent dat we bepaalde verhoudingen moeten hebben tussen onze hoeksnelheden. Deze vergelijking is daar zeer handig voor, we kunnen onze gewenste hoeksnelheid invullen in x. Als we dit dan uitrekenen, bekomen we de PWM-waarde die we doorschrijven naar de servomotor om die snelheid te krijgen. Deze rechte heeft wel een soort van einde aangezien de servomotor een minimale snelheid en een maximale snelheid heeft.

6. Programma's

6.1 Transmitter

```
#include <SPI.h>
#include <RF24.h> //bibliotheken om de NRF24 aan te sturen en binnen te lezen
#include <Servo.h> //bibliotheek om de servo aan te sturen

#define Blauw 4
#define Rood 2 //pinnen voor drukknoppen
#define Pulse 7 //pin om puls binnen te lezen

#define ROLE_TX true //Rol instellen als transmitter

#define SPI_CSN 10
#define SPI_CE 9
#define NRF_IRQ 3 //CSN, CE en IRQ pinnen instellen
int teller = 0;

Servo hoofdMotor; //de hoofmotor een naam geven om aan te sturen

bool Knop_Rood;
bool Knop_Blauw; //statussen van de drukknoppen

const byte address[6] = {0x2B, 0x96, 0x09, 0xB6, 0x35};
RF24 radio(SPI_CE, SPI_CSN);

struct packedFrame{//Pakket dat we doorsturen
    int16_t PWMS; //PWM-waarde die naar de servo moet doorgeschreven worden bij
de receiver
    bool Knop_Blauw;
    bool Knop_Rood; //statussen van de drukknoppen die we zullen doorsturen
};

#if defined(ROLE_TX) && (ROLE_TX == true)//Kijken of de rol transmitter of
receiver is, dit wordt uitgevoerd bij de transmitter
    packedFrame txBuff;
    uint8_t rxBuff[32];
#else
    packedFrame rxBuff;
#endif

bool newDataFromISR = false;

#if !defined(ROLE_TX) || (ROLE_TX == false)//dit zal het programma enkel
uitvoeren als de rol receiver is
    void isrNRF(void){
```

```
bool tx, fail, rx;
radio.whatHappened(tx, fail, rx);
if(rx){
    radio.read(&rxBuff, radio.getDynamicPayloadSize());
    radio.writeAckPayload(1, &rxBuff.lights, 1);
    newDataFromISR = true;
}
}
#endif

void setup() {
    pinMode(Blauw, INPUT_PULLUP);
    pinMode(Rood, INPUT_PULLUP);
    pinMode(Pulse, INPUT_PULLUP); //in te lezen pin en de pinnen van de
    drukknooppn als input instellen

    hoofdMotor.attach(6); //Pin motor
    Serial.begin(115200); //Seriële monitor instellen
    Serial.println("NRF24 on Arduino with structures");
    #if defined(ROLE_TX) && (ROLE_TX == true)
        Serial.println("\tRole = transmitter");
    #else
        Serial.println("\tRole = receiver");
    #endif
    if(radio.begin()){
        if(radio.isChipConnected()){ //kijken of de chip verbonden is
            if(radio.isPVariant()){ //Hier worden alle instellen gedaan voor
                goede communicatie
                    Serial.print("NRF:\tPro ");
                    radio.enableDynamicPayloads();
                    radio.enableAckPayload();
                    radio.setDataRate(RF24_250KBPS); //hoe snel hij zal versturen
                }else{
                    Serial.print("NRF:\tBasic ");
                    radio.setDataRate(RF24_1MBPS);
                }
            Serial.println("radio detected");
            radio.setPALevel(RF24_PA_MAX);
            radio.setChannel(100); //aantal kanalen
            radio.setRetries(7,3); //aantal retries en de delay er tussen
        }else{
            Serial.println("ERR:\tNo radio detected!");
            while(1); //no need to continue
        }
    }else{
        Serial.println("ERR:\tFailed to create RF24 object");
        while(1); //no need to continue
    }
}
```

```
#if defined(ROLE_TX) && (ROLE_TX == true)
    radio.stopListening();
    radio.openWritingPipe(address);
#else
    radio.openReadingPipe(1, address);
    radio.startListening();
    radio.maskIRQ(1,1,0);
    pinMode(NRF_IRQ, INPUT_PULLUP); //interrupt wordt enkel in de receiver
gebruikt
    attachInterrupt(digitalPinToInterrupt(NRF_IRQ), isrNRF, FALLING);
#endif
}

uint8_t cnt = 0;

void loop() {
    #if defined(ROLE_TX) && (ROLE_TX == true)
        txBuff.Knop_Blauw = digitalRead(Blauw);
        txBuff.Knop_Rood = digitalRead(Rood); //statussen knoppen inlezen om dan
later door te sturen
        if(!digitalRead(Blauw)){
            hoofdMotor.writeMicroseconds(950); //motor die beweegt als blauw
ingedrukt is geweest
        }
        if(!digitalRead(Rood)){
            hoofdMotor.writeMicroseconds(0); //motor staat stil als rood
ingedrukt is geweest
        }

        txBuff.PWMS = 2705; //PWM-waarde servomotor instellen om door te
sturen

        if(radio.write(&txBuff, sizeof(txBuff))){
            if(radio.isAckPayloadAvailable()){
                radio.read(&rxBuff, radio.getDynamicPayloadSize());
                teller = 0; //teller beveiliging 0 maken
            }else{
                Serial.println("NRF:\tTX OK, no piggyback data");
            }
            cnt++;
        }else{
            Serial.println("NRF:\tTX failed!");
            //Stukje code dat de motor zal stoppen als er geen communicatie meer
is

            teller += 1;
            if (teller == 10){
                hoofdMotor.writeMicroseconds(0);
            }
        }
    }
}
```

```
    }  
#endif  
}
```

6.2 Receiver

```
#include <SPI.h>  
#include <RF24.h> //bibliotheken om de NRF24 aan te sturen en te laten  
verzenden  
//hier is geen bibliotheek voor de servo nodig omdat we met registers werken  
  
int currentMillis = 0; //een variabele om de timer mee te vergelijken  
  
#define Analog A0 //analoge pin voor de potentiometer  
#define servoPin 6 //Pin voor de servo  
  
#define ROLE_TX false //rol instellen als receiver  
  
#define SPI_CSN 10  
#define SPI_CE 9  
#define NRF_IRQ 3 //pinnen van de NRF24 instellen  
  
const byte address[6] = {0x2B, 0x96, 0x09, 0xB6, 0x35}; //nodig om NRF24 aan  
te sturen  
bool var = false; //controleer variabele om de servo aan te sturen  
  
uint16_t tijdvoorregister = 0; //waarde om servo mee aan te sturen  
  
RF24 radio(SPI_CE, SPI_CSN); //pinnen om mee te communiceren instellen  
  
struct packedFrame{  
    int16_t PWMS;  
    bool Knop_Blauw;  
    bool Knop_Rood;  
}; //pakket die de receiver verwacht te ontvangen  
  
#if defined(ROLE_TX) && (ROLE_TX == true) //kijken of de rol transmitter is  
    packedFrame txBuff;  
    uint8_t rxBuff[32];  
#else  
    packedFrame rxBuff;  
#endif  
  
bool newDataFromISR = false;  
  
#if !defined(ROLE_TX) || (ROLE_TX == false) //kijken of de rol receiver is
```

```
void isrNRF(void){
  bool tx, fail, rx;
  radio.whatHappened(tx, fail, rx);
  if(rx){
    radio.read(&rxBuff, radio.getDynamicPayloadSize());
    newDataFromISR = true;
  }
}
#endif

void setup() {
  pinMode(servoPin, OUTPUT); //Servopin instellen als output
  tijdvoorregister = 2975; //komt overeen met stilstand van de servo

  cli(); //Schakelt interrupts uit zodat de ISR zelf niet 'geïnterrupt' wordt

  TCCR1A = 0; //timer counter control 1 register A instellen

  TCCR1B &= 0xD0;

  TCCR1B |= 0x2; //timer counter control 1 register B instellen

  TIMSK1 |= 0X02; //timer 1 instellen

  OCR1A = 16999; //referentiewaarde voor de timer

  TCNT1 = 0; //timer die omhoog zal tellen
  //de laatste 2 worden vergeleken met elkaar, als ze gelijk zijn, wordt de
  interrupt ingeschakeld

  sei(); //schakelt interrupts terug in

  Serial.begin(115200);
  Serial.println("NRF24 on Arduino with structures");
  Serial.println("\tRole = receiver");
  if(radio.begin()){
    if(radio.isChipConnected()){
      if(radio.isPVariant()){
        Serial.print("NRF:\tPro ");
        radio.enableDynamicPayloads();
        radio.enableAckPayload();
        radio.setDataRate(RF24_250KBPS); //snelheid om mee door te sturen
      }else{
        Serial.print("NRF:\tBasic ");
        radio.setDataRate(RF24_1MBPS);
      }
    }
    Serial.println("radio detected");
    radio.setPALevel(RF24_PA_LOW);
  }
}
```

```
    radio.setChannel(100); //aantal kanalen
    radio.setRetries(7,3); //aantal retries en delay er tussen
}else{
    Serial.println("ERR:\tNo radio detected!");
    while(1);
}
}
}else{
    Serial.println("ERR:\tFailed to create RF24 object");
    while(1);
}
}
#if defined(ROLE_TX) && (ROLE_TX == true) //kijken of de rol transmitter is
    radio.stopListening();
    radio.openWritingPipe(address);
#else
    radio.openReadingPipe(1, address);
    radio.startListening();
    radio.maskIRQ(1,1,0);
    pinMode(NRF_IRQ, INPUT_PULLUP);
    attachInterrupt(digitalPinToInterrupt(NRF_IRQ), isrNRF, FALLING);
//Interrupt instellen
#endif
}

uint8_t cnt = 0;

int16_t testCmp;

void loop() {
    if(newDataFromISR){ //'als hij nieuwe data krijgt'
        if (currentMillis - millis() > 5000){
            tijdvoorregister = map(analogRead(Analog), 0, 1023, 2800, 2950);
        }
        int currentMillis = millis();//deze 4 lijntjes dienen als beveiliging
        moest de communicatie wegvallen als de tijd sinds de laatste transmissie 5s is,
        stopt de servo

        if(!rxBuff.Knop_Blauw){ //rxBuff is de waarde die hij doorgestuurd
        krijgt binnenlezen
            var = true; //opslaan of hij moet draaien of niet
        }
        else if(!rxBuff.Knop_Rood){
            var = false; //opslaan of hij moet draaien of niet
        }
        if (!var){
            tijdvoorregister = map(analogRead(Analog), 0, 1023, 2800,
            2950); //stilstand is aanpasbaar met een potentiometer omdat de waarde
            temperatuurafhankelijk is
        }
    }
}
```



```
    else if (var){
        tijdvoorregister = rxBuff.PWMS; //waarde die de servo moet krijgen
doorsturen
    }
    newDataFromISR = false;
}
}

ISR(TIMER1_COMPA_vect){
    bool Status = digitalRead(servoPin); //kijken of er een 1 of een nul naar
de servo doorgestuurd wordt
    if(!Status){ //als er een 0 wordt doorgestuurd
        digitalWrite(servoPin, HIGH);
        OCR1A = tijdvoorregister; //er wordt dus een nieuwe tijd ingesteld: als
deze weer gelijk zijn wordt de interrupt aangeroepen en wordt er weer een lage
waarde voor 16999 microseconden doorgestuurd
    }
    else{
        digitalWrite(servoPin, LOW);
        OCR1A = 16999;
    }
    TCNT1 = 0; //timer wordt weer op 0 gezet
}
```

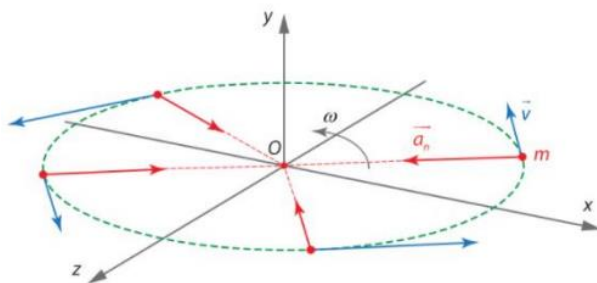
7. Wetenschappelijke studie

7.1 Meten van krachten

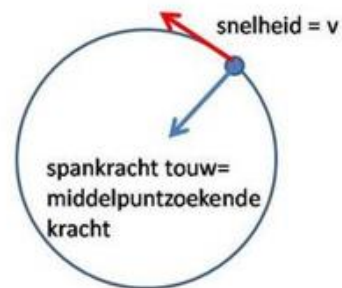
7.1.1 Theorie

7.1.1.1 Centripetale of middelpuntzoekende kracht

De centripetale of middelpuntzoekende kracht is de kracht die een voorwerp in een cirkel doet bewegen, dit gebeurt als er een kracht met een constante grootte telkens haaks staat op de bewegingsrichting. Hierbij wijst de kracht continu naar het middelpunt. Deze kracht ontstaat omdat er op elk ronddraaiend lichaam een normaalversnelling \vec{a}_n werkt. De normaalversnelling doet de snelheidsvector voortdurend van richting veranderen. Door de normaalversnelling ontstaat de centripetale of middelpuntzoekende kracht.



Figuur 49 : Centripetale kracht



Figuur 50 : Centripetale kracht

- **Grootte:**

$$F_c = \frac{m * v^2}{r}$$

$$F_c = m \cdot a_n$$

$$F_c = m * r * \omega^2$$

- **Richting:**

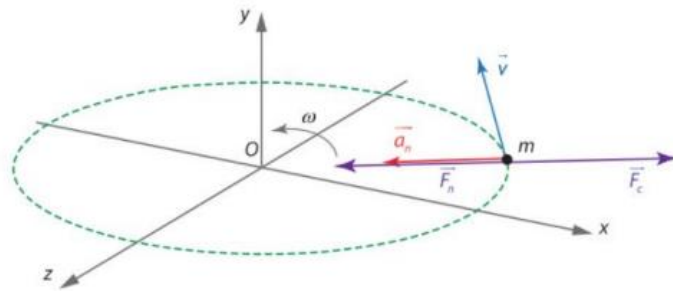
De centripetale kracht heeft dezelfde richting als de normaalversnelling

- **Zin:**

De zin is naar het middelpunt gericht, hetzelfde als de normaalversnelling.

7.1.1.2 Centrifugale of middelpuntvliedende kracht

De centrifugaal of middelpuntvliedende kracht \vec{F}_c is een schijnkracht die wordt waargenomen door een zelf meebewegende waarnemer, het zorgt ervoor dat het lichaam niet wegvliegt. De centrifugaalkracht is de traagheidsvector van een ronddraaiende beweging, elke bewegingsverandering wordt tegengewerkt door een traagheidsvector.



Figuur 51 : centrifugale en centripetale kracht



Figuur 52 : centrifugale en centripetale kracht

- **Grootte:**

$$F_C = \frac{m * v^2}{r}$$

$$F_c = m * r * \omega^2$$

$$F_c = -F_N$$

- **Richting:**

Volgens de straal van de cirkelbaan

- **Zin:**

Tegengesteld aan de centripetale kracht, dus naar buiten gericht.

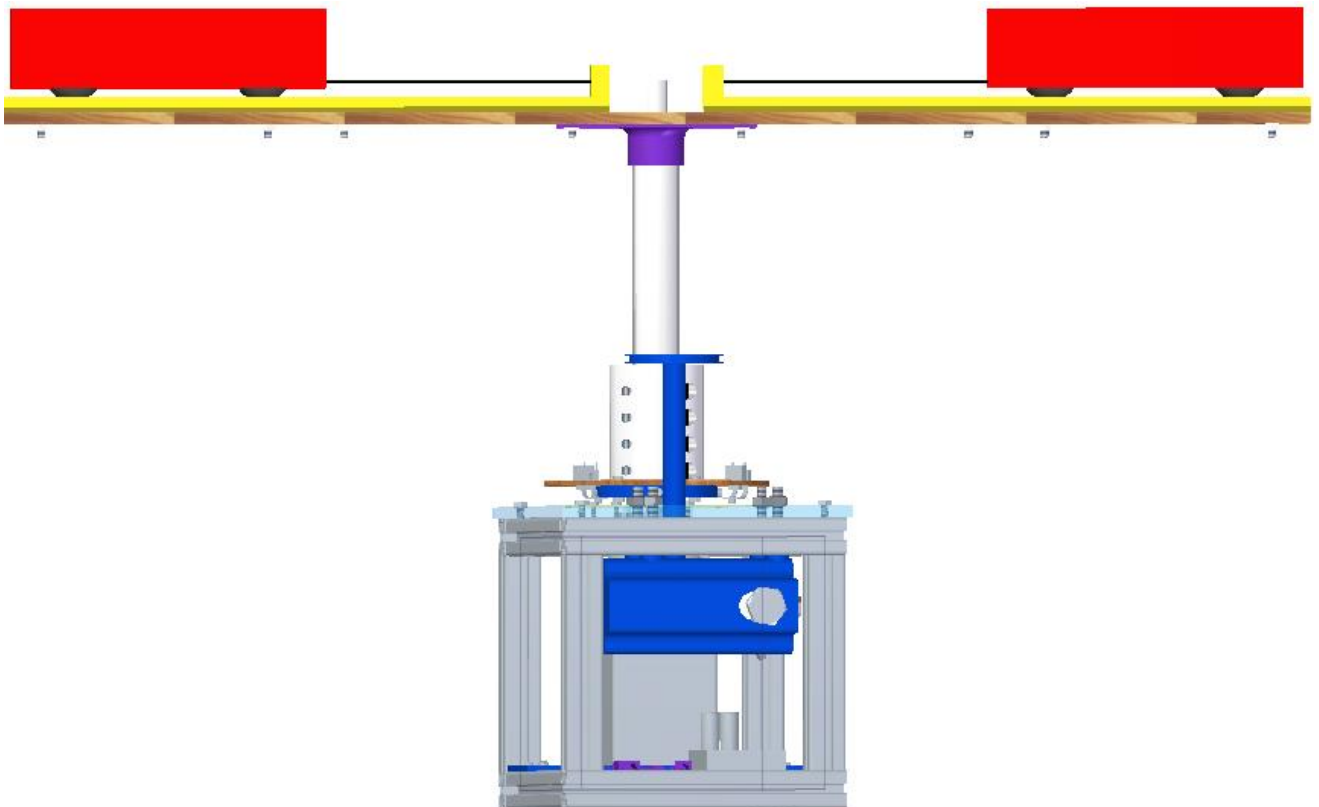
7.1.2 Praktijk

7.1.2.1 Meetopstelling met karretjes

Met onze polyp willen we graag de middelpuntvliedende kracht aantonen. Hierop hebben we dan ook verschillende metingen uitgevoerd, dit met de opstelling waarbij we gebruik maken van sensoren die uitgewerkt zijn in de vorm van karretjes.

1) Opstelling

Om de krachtmetingen uit te voeren gaan we te werk met de PASCO wireless smart cart (zie ook 8.4.1). In deze karretjes zit dus een krachtsensor, aan deze sensor kunnen we haakjes bevestigen, en aan die haakjes hangen we een touwtje. Dit touwtje is met de ene zijde verbonden met de karretjes en aan de andere zijde met een bevestigingsstuk dicht bij de as. Ook staan de karretjes op railsporen waarop deze kunnen rijden (deze zijn geconstrueerd voor de achtbaan van vorig jaar). Het gevolg is dat door de versnelling \vec{a}_n van de rotatiebeweging de karretjes de neiging hebben om naar buiten te bewegen, hierbij zal de koord gespannen raken en zo zullen we dus de spankracht in de koorde kunnen meten.



Figuur 53 : opstelling meten van krachten

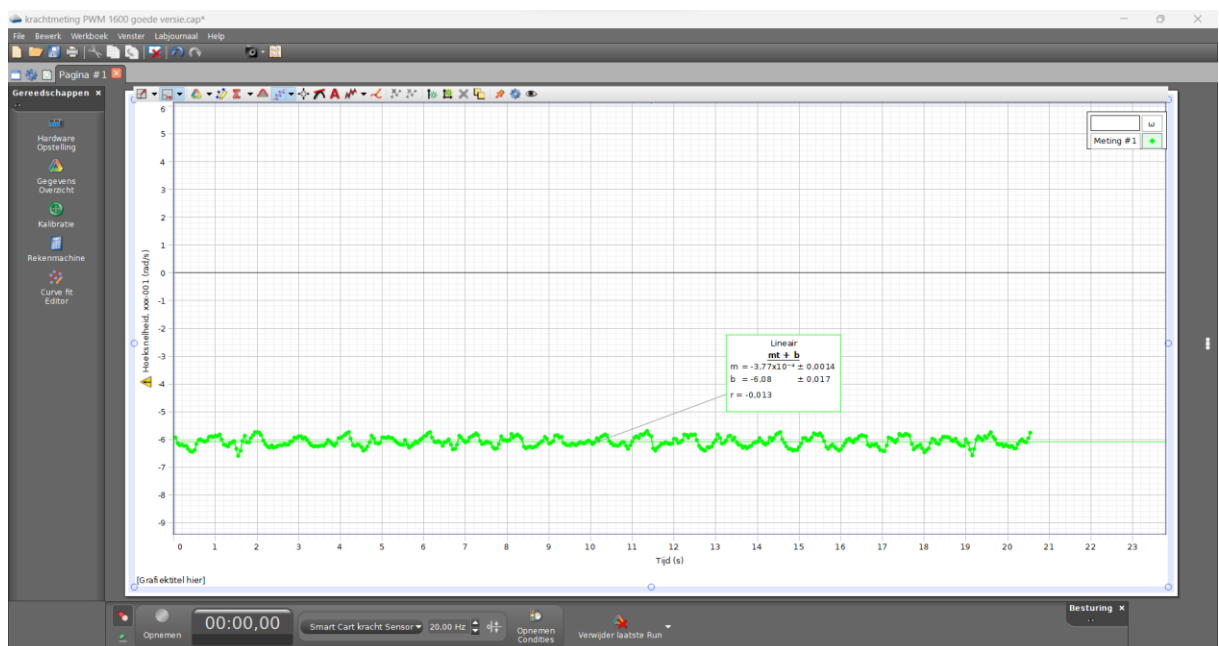
2) Hoe zijn we te werk gegaan?

We hebben meerdere metingen uitgevoerd bij verschillende PWM-waarden, dit resulteerde in verschillende rotatiesnelheden. Bij deze verschillende rotatiesnelheden hebben we verschillende krachten gemeten, we hebben dit negen keer gedaan. Al deze waarden hebben we dan in een kracht-snelheidsgrafiek geplaatst, waaruit we uiteindelijk een tweedegraadsvergelijking haalden. Dit klopt ook want we zien deze tweedegraadsfunctie ook terug in de formule van de middelpuntvliedende kracht: $\vec{F}_C = m * r * \omega^2$

7.1.2.1.1 Meetresultaten

1) Voorbeeld meting bij een PWM van 1600

Het meten van de rotatiesnelheid:



Figuur 54 meten van rotatiesnelheid (PASCO)

De rotatiesnelheid meten we aan de hand van de rotatiesensor van PASCO. Bij elke meting hebben we een tijdsinterval van ongeveer twintig seconden genomen.

Uit de meting konden we telkens een vergelijking van de vorm $f(t) = mt + b$ halen. Hier in het voorbeeld hebben we de vergelijking: $f(t) = f(t) = -3,77 \cdot 10^{-4}t - 6,08$

Hieruit kunnen we besluiten dat de t-waarde in de vergelijking te verwaarlozen is en dat de hoeksnelheid dus gelijk is aan 6,08 rad/s. Het minteken in de vergelijking komt doordat we omgekeerd meten met de rotatiesensor, in de constructie hebben we geen rekening gehouden met de positieve draaiing van de sensor. Dit is dan ook te zien in de grafiek.

Het meten van de kracht:



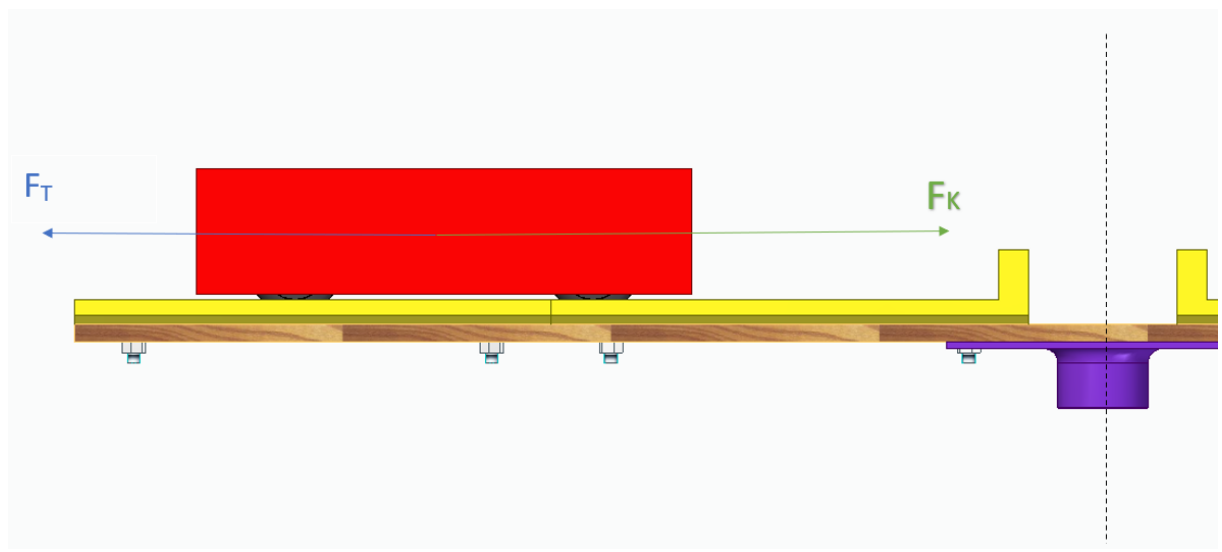
Figuur 55 : meten van centripetale kracht (PASCO)

De kracht hebben we gemeten aan de hand van een krachtsensor. Opnieuw hebben we per meting een tijdsinterval genomen van ongeveer twintig seconden.

Uit de meting konden we opnieuw een vergelijking van de vorm $f(t) = mt + b$ halen. Hier in het voorbeeld hebben we de vergelijking: $f(t) = 5,07 \cdot 10^{-4}t + 2,65$.

We zien dat de t-waarde terug te verwaarlozen is en dat we de gemeten spankracht in het touw dus kunnen gelijkstellen aan 2,65N.

Het berekenen van de kracht aan de hand van de rotatiesnelheid:



Figuur 56 : schematische voorstelling oefening

- F_k is hier de kracht in de koorde = middelpuntzoekende kracht
- F_T is de tangentiële kracht = middelpuntvliedende kracht
- m is hier de massa van het karretje = 0,253 kg
- r is hier de straal van de opstelling = 0,248 m

De som van de krachten op de X-as:

$$\begin{aligned}\sum \vec{F}_x: \vec{F}_k + \vec{F}_T &= 0 \\ F_k &= F_T \\ m \cdot r \cdot \omega^2 &= F_n \\ 0,253 \text{ kg} \cdot 0,248 \text{ m} \cdot 6,08^2 \text{ rad/s} &= F_k \\ F_k &= 2,32 \text{ N}\end{aligned}$$

Besluit

Na deze eerste meting en de berekende versie ervan kunnen we al een duidelijk besluit stellen: 2,65 N \neq 2,32 N. We bekomen dus dat de gemeten kracht groter is dan de theoretisch berekende kracht. Eerst dachten we dat dit verschil te danken was aan de weerstand tussen de karretjes en de rails, maar dit zou niet overeenstemmen met de realiteit, want in de realiteit zou de gemeten kracht juist kleiner moeten zijn dan de berekende kracht, omdat je natuurlijk wrijvingsverliezen hebt die de krachten zullen tegenwerken. We zitten hier met een krachtverschil van ongeveer 0,33 N.

We moesten dus op zoek naar een andere oorzaak voor het relatief grote verschil in de berekende en gemeten krachten. Deze oorzaak vonden we dan ook en was te danken aan de krachtsensor.

Met onze krachtsensor hebben we een meting uitgevoerd waarbij er geen massa aan de sensor hing, dit zou eigenlijk het nulpunt van de sensor moeten zijn. Tot onze verbazing was dit niet zo. Als we dit te weten kwamen, hebben we dan ook meteen een PASCO-expert geraadpleegd, en deze zei dat we het toestel moesten herkalibreren naar de fabrieksinstellingen. Dit deden we dan ook, de krachtsensor gaf bij geen belasting een waarde van 0,37 Newton weer.

Het probleem hebben we dan ook simpel opgelost door de gemeten kracht te verminderen met die 0,37 Newton. Zo kwamen we zeer mooie waarden uit die op enkele honderdsten na overeenstemden met de berekende waarden (zie tabel hieronder).

2) Alle meetresultaten

De metingen zelf hebben we tien keer uitgevoerd bij verschillende PWM-signalen, wat resulteerde in verschillende rotatiesnelheden.

We zien in bovenstaande tabel, in de eerste kolom de PWM-signalen. Deze signalen zijn aan te leggen via het programma (Arduino). In de tweede kolom zien we de bijhorende rotatiesnelheden. In de derde kolom de gemeten middelpuntzoekende kracht, en in de vierde kolom de gemeten middelpuntzoekende kracht verminderd met 0,37 N. Ten slotte hebben we in de laatste kolom de berekende waarden voor de kracht bij die rotatiesnelheid.

PWM-sigitaal	Rotatiesnelheid in radialen per seconde (rad/s)	Middelpuntzoekende kracht in newton (N)	Middelpuntzoekende kracht in newton (N) verminderd met 0,37 N	Middelpuntzoekende kracht in newton (N) (berekend)
800	0,795	0,403	0,033	0,0397
900	2	0,628	0,258	0,251
1000	2,74	0,811	0,441	0,472
1100	3,39	1,1	0,73	0,722
1200	4,05	1,36	0,99	1,03
1300	4,56	1,67	1,3	1,31
1400	5,17	2	1,63	1,68
1500	5,64	2,34	1,97	2,00
1600	6,08	2,65	2,28	2,32
1700	6,49	3	2,63	2,65

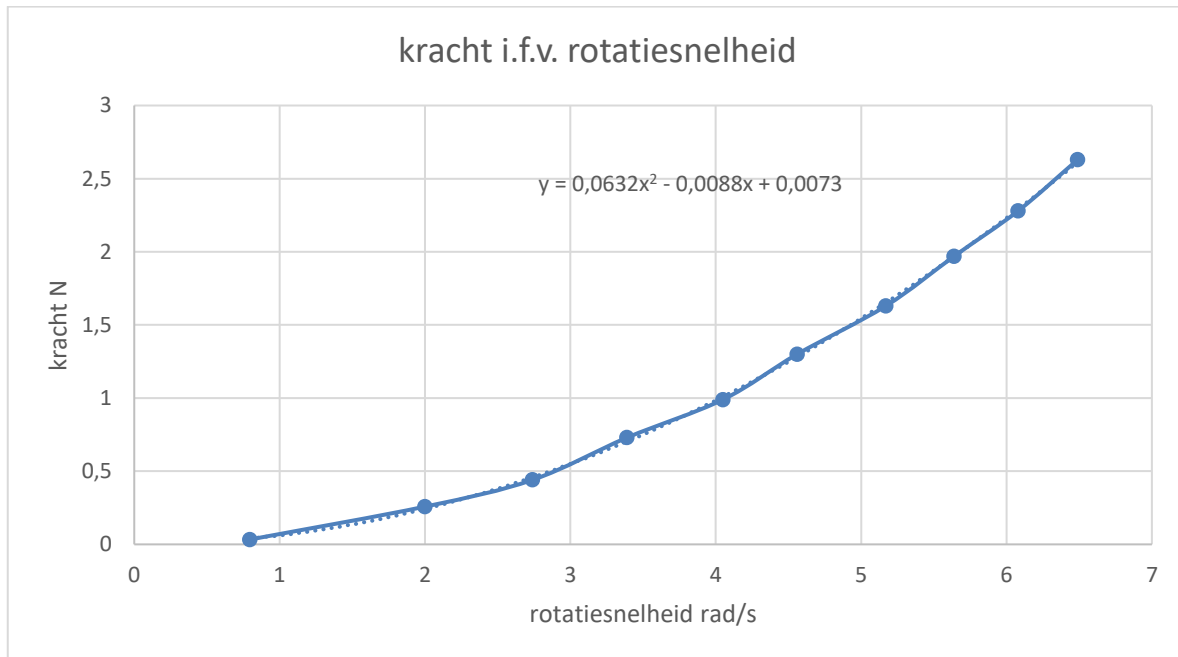
Besluit:

Uit de tabel kunnen we nu besluiten dat de gemeten krachten en de berekende krachten nagenoeg gelijk zijn aan elkaar. De verschillen zijn hoogstwaarschijnlijk aan meetfouten te danken, we hebben dan ook verschillen die niet groter zijn dan 0,1 newton. Ook zien we dat de verschillen groter zijn bij lage rotatiesnelheden, dit zal hoogstwaarschijnlijk komen omdat de motor niet geschikt is om op zulke lage toerentallen te draaien, en daardoor wat begon te schokken.

3) Verwerking van meetresultaten

Toen we alle metingen hadden verzameld, hebben we deze uitgezet in een snelheids-kracht-grafiek. Hierbij hebben we de snelheid op de X-as geplaatst en de kracht op de Y-as. Het is de bedoeling om een tweedegraadsfunctie te bekomen, wat we kunnen verklaren door de formule van de middelpuntvliedende kracht: $\vec{F}_c = m * r * \omega^2$

Grafiek van de gemeten waarden



Figuur 57 : kracht i.f.v. rotatiesnelheid

Hier zien we de grafiek die we bekomen met de gemeten krachtwaarden. We konden uit de metingen een vergelijking van de vorm $f(x) = Ax^2 + Bx + C$ halen. Hier in het voorbeeld hebben we de vergelijking: $f(s) = 0,0632s^2 - 0,0088s + 0,0073$

Als we naar de formule kijken van de middelpuntvliedende kracht moeten we in principe een vergelijking van de vorm $f(x) = Ax^2$ verkrijgen, waarbij de A-waarde theoretisch overeenkomt met het product van massa en straal. We zien dan ook in de bekomen formule dat we de B en C-waarde kunnen verwaarlozen. Het is logisch dat de C waarde moet gelijk zijn aan nul, want als je een rotatiesnelheid van nul rad/s hebt, moet je ook een kracht van nul newton uitkomen.

Als we nu deze B- en C-waarde verwaarlozen, dan kunnen we een formule opstellen voor de centripetale kracht, die overeenstemt met de theoretische vorm.

We hebben dus: $f(s) = 0,0632s^2$

Delen we nu de A-waarde door de opgemeten straal, dan bekomen we de massa.

$$\frac{0,0632}{0,248} = 0,255 \text{ kg}$$

De massa van het karretje is gelijk aan 0,253 kg, dus we zitten hier met een zeer nauwkeurig eindresultaat.

Ook kunnen we nu de exacte versie van de grafiek zien in volgend puntje, dit is dan de grafiek met de berekende waarden.

7.2 Meten van rotatiesnelheid

7.2.1 Theorie

7.2.1.1 Eenparig cirkelvormige beweging (ECB)

Een punt of lichaam beschrijft een cirkelvormige beweging als dat punt of lichaam in dezelfde zin op een cirkelbaan beweegt, en daarbij in gelijke tijden gelijke booglengten aflegt.

De vectoriële grootheden

1) De normaalversnelling \vec{a}_n :

Door een voortdurende wijziging van de richting en de omtreksnelheidsvector treedt de normaalversnelling op.

- **Grootte:**

$$a_n = \frac{v^2}{r}$$

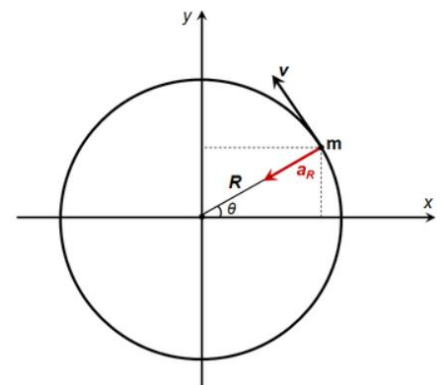
$$a_n = r \cdot \omega^2$$

- **Richting:**

Richting van de straal die bij het punt van de beweging hoort.

- **Zin:**

Naar het middelpunt van de cirkel.



Figuur 58 : normaalversnelling

2) De hoeksnelheid $\vec{\omega}$:

Deze geeft de hoekverandering per tijdseenheid aan.

- **Grootte:**

Verhouding van de doorlopen hoek en de tijd nodig om deze hoek te doorlopen.

$$\omega = \frac{\Delta\theta}{\Delta t}$$

- **Richting:**

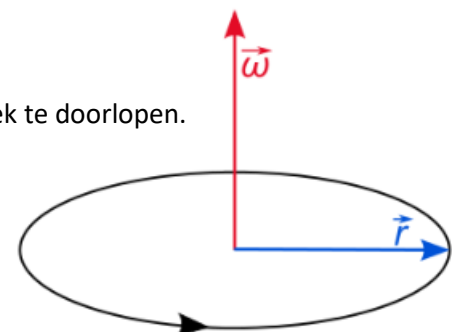
Loodrecht op het vlak van de cirkelvormige beweging.

- **Zin:**

Bepalen door rechterhandregel.

- **Aangrijpingspunt:**

Middelpunt van de cirkelvormige beweging.



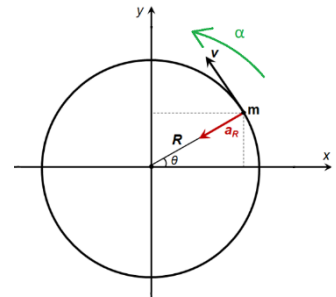
Figuur 59 : hoeksnelheid

7.2.1.2 Eenparig veranderlijke cirkelvormige beweging (EVCB)

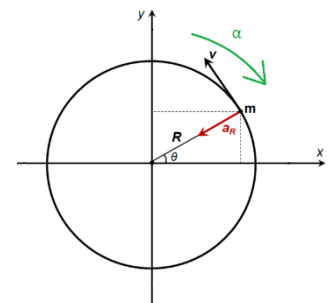
Dit is een beweging die volgens een cirkelvormige baan verloopt en waarbij de hoeksnelheid per tijdsinterval, hoe klein ook genomen, met een constante waarde toe- of afneemt. De verandering van de hoeksnelheid per tijdsinterval is constant.

We hebben hierbij 2 soorten:

- De eenparig versnelde cirkelvormige beweging, hierbij hebben we een constante toename van de hoeksnelheid in gelijke tijdsintervallen, de hoeksnelheid en de hoekversnelling hebben hierbij eenzelfde zin.
- De eenparig vertraagde cirkelvormige beweging, hierbij hebben we een constante afname van de hoeksnelheid in gelijke tijdsintervallen. De hoeksnelheid en de hoekversnelling hebben hierbij een tegengestelde zin.



Figuur 60 : versnelling



Figuur 61 : vertraging

De vectoriële grootheden

- 1) De hoekversnelling $\vec{\alpha}$:

Dit is de verhouding van de verandering van de hoeksnelheid en de verandering van de tijd.

Grootte:

$$\alpha = \frac{\Delta\omega}{\Delta t}$$

Richting:

Loodrecht op het vlak waarin de beweging zich afspeelt.

Zin:

Bij een versnelde beweging heeft hij dezelfde zin als de hoeksnelheid ω , bij een vertraagde beweging heeft hij een tegengestelde zin als de hoeksnelheid ω .

- 2) De hoeksnelheid (zie 7.2.2)

Vergelijkingen:

- 1) De doorlopen hoek (= opp. onder (t; ω)-assenstelsel):

$$\theta = \theta_0 + \omega_0 \cdot t + \frac{\alpha \cdot t^2}{2}$$

- 2) Hoeksnelheid op bepaald ogenblik t:

$$\omega_t = \omega_0 + \alpha \cdot t$$

- 3) Omtreksnelheid v:

$$v = \omega \cdot r$$

7.2.2 Praktijk

We hebben een meting uitgevoerd waarin de motor versnelt vanuit stilstand, daarna hebben we de motor laten draaien op een constante snelheid, ten slotte hebben we hem laten vertragen tot stilstand.

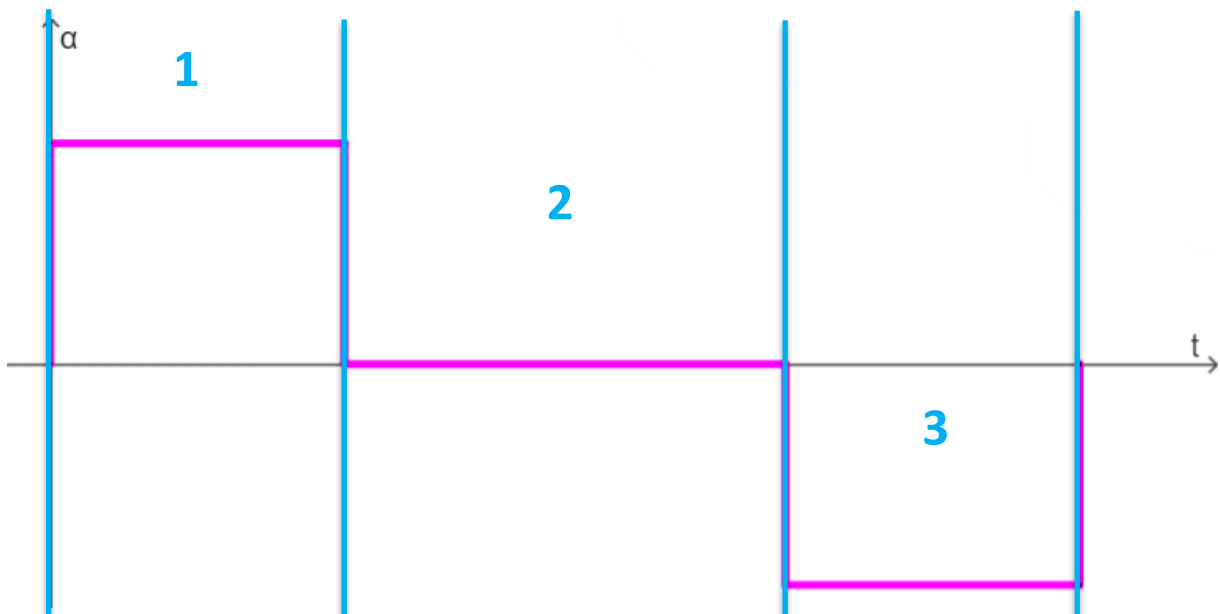
We hebben dit gedaan door een PWM-waarde van 1500 in het programma te stoppen, dit resulteert dan in een bepaalde hoeksnelheid.

Hieronder zullen we de verschillende grafieken bespreken van de hoeksnelheid, de hoekversnelling en de doorlopen hoek.

Een probleem in de opstelling is dat we geen rekening gehouden hebben met de meetzin van onze rotatiesensor, dit resulteert dan ook in tegengestelde resultaten van hoeksnelheid, doorlopen hoek en hoekversnelling.

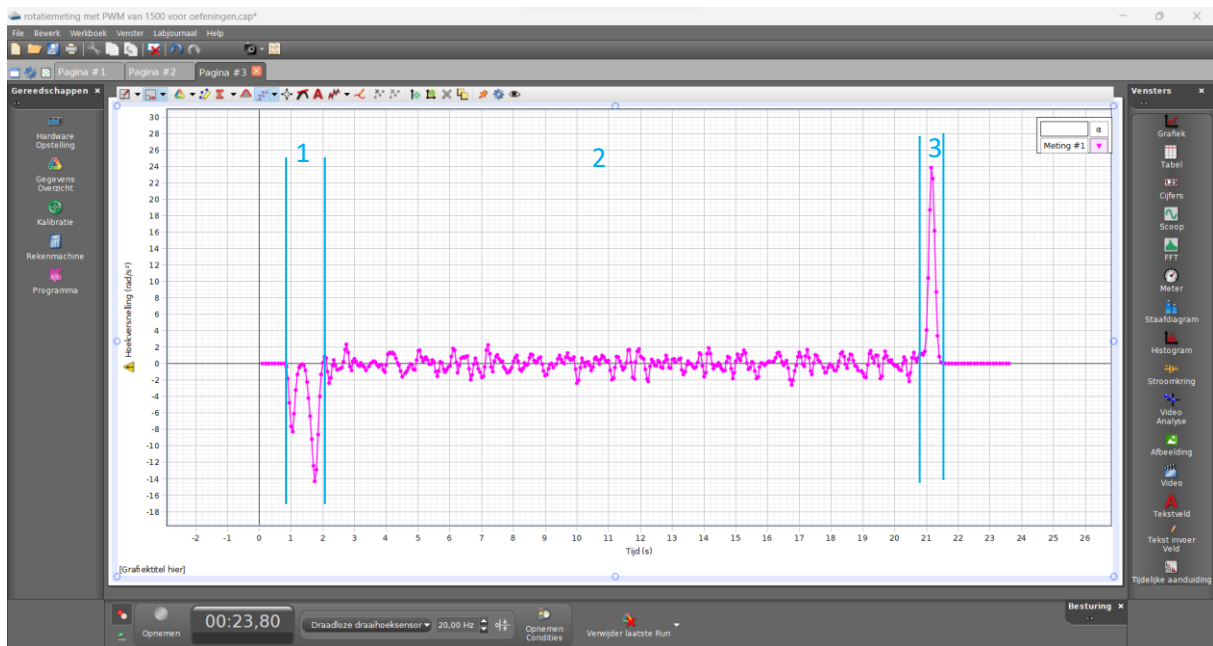
1) Hoekversnelling i.f.v. de tijd

Als we uitgaan van een eenparig veranderlijke cirkelvormige beweging bij het begin en het einde van de beweging dan kunnen we zeggen dat de hoekversnelling α in functie van de tijd er als volgt moet uitzien:



Figuur 62 : verwachte grafiek van hoekversnelling i.f.v. de tijd

Als we dit vergelijken met onze bekomen grafiek van de meting kunnen we besluiten dat deze niet lopen zoals verwacht:



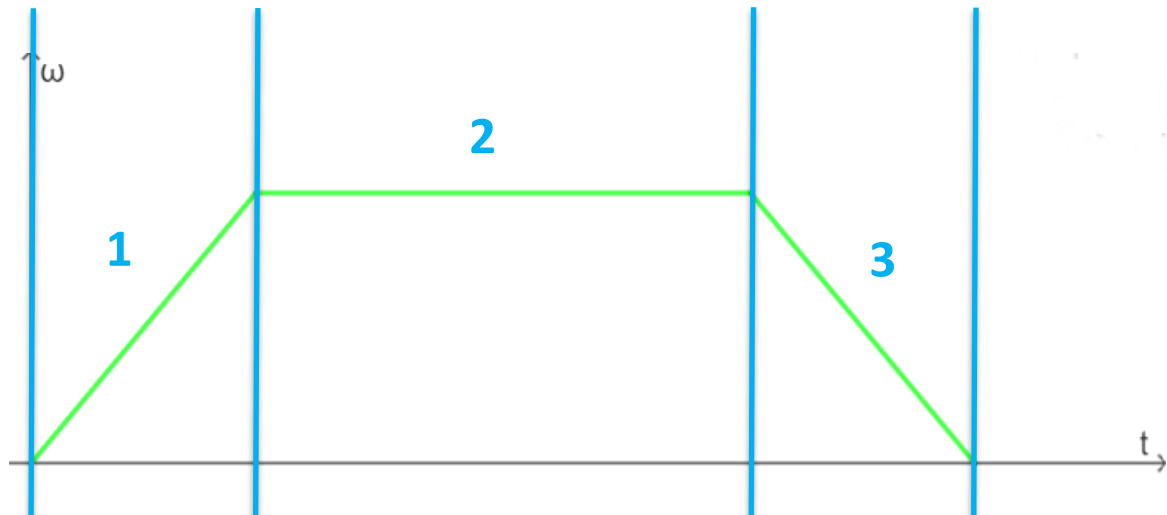
Figuur 63 : gemeten hoekversnelling i.f.v. de tijd (PASCO)

We kunnen deze grafiek opdelen in drie delen:

1. Het eerste deel is de versnelling, we kunnen zeggen dat dit geen constante lijn is als hierboven. Het besluit is dus dat we hier niet met een eenparig versnelde cirkelvormig beweging zitten, maar met een versnelde beweging, waarbij de versnelling dus niet constant is. Het gevolg is dan ook dat de formules van de EVCB hier niet gelden en wij hier geen berekeningen kunnen op uitvoeren.
2. In het tweede deel zien we dat de gemiddeld genomen versnelling gelijk is aan nul. Dit is ook logisch want we zitten hier met een constante hoeksnelheid.
3. In het derde en laatste deel zien we dat we met een vertraging zitten. We kunnen opnieuw concluderen dat dit geen constante is en dus ook geen eenparig vertraagde cirkelvormige beweging, maar een vertraagde beweging, waarbij de versnelling dus opnieuw niet constant is.

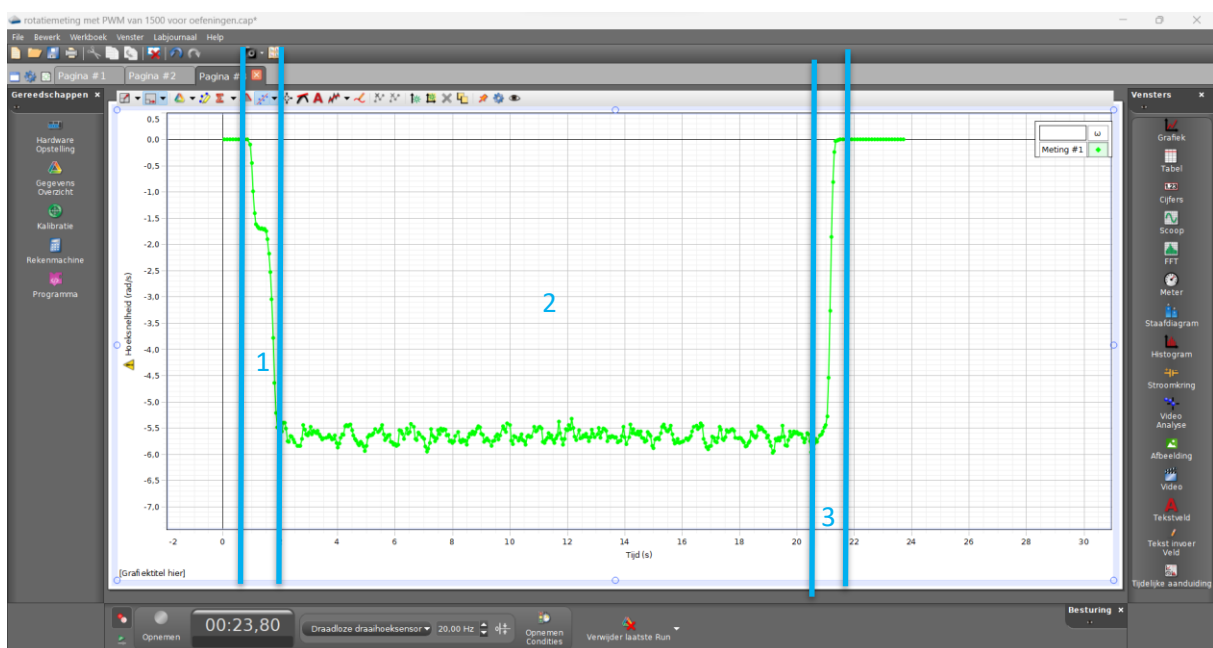
2) De hoeksnelheid i.f.v. de tijd

Als we uitgaan van een eenparig versnelde cirkelvormige beweging dan kunnen we zeggen dat de hoeksnelheid ω in functie van de tijd er als volgt moet uitzien:



Figuur 64 : verwachte grafiek van hoeksnelheid i.f.v. de tijd

Hier zien we het verschil minder duidelijk dan bij de grafiek met de hoekversnelling, maar we kunnen opnieuw hetzelfde besluiten.



Figuur 65 : gemeten hoeksnelheid i.f.v. de tijd (PASCO)

We kunnen deze grafiek opnieuw opdelen in drie delen:

1. Het eerste deel is de versnelling. Op het eerste gezicht lijkt dit op een constante toename van de hoeksnelheid in functie van de tijd, maar als we deze van naderbij bekijken, kunnen we opnieuw concluderen dat dit geen eenparig versnelde cirkelvormige beweging is, maar een versnelde cirkelvormige beweging. We kunnen dit zien doordat de meetpunten in de versnelling niet op éénzelfde lijn liggen.
2. In het tweede deel zien we dat we een constante hoeksnelheid ω bekomen, hier kunnen we dus wel zeggen dat we in een eenparig cirkelvormige beweging zitten.
3. In het derde en laatste deel zien we dat we met een vertraging zitten. We kunnen opnieuw concluderen dat dit geen eenparig vertraagde cirkelvormige beweging is, maar een vertraagde beweging. We kunnen dit zien doordat de meetpunten in de vertraging niet op éénzelfde lijn liggen.

Bepalen van de hoeksnelheid in het lineaire gedeelte (deel 2):

Uit deze grafiek kunnen we nu wel in het lineaire gedeelte gebruik maken van de formules die we geleerd hebben in de tweede graad.

We kunnen dus uit het middelste deel een lineaire vergelijking halen van de vorm $y(t) = mt + b$

In ons voorbeeld is deze: $y(t) = -0,00150t - 5,64$.

Hieruit kunnen we dus een hoeksnelheid ω nemen van 5,64 rad/s.

Met deze snelheid kunnen we vervolgens de doorlopen hoek berekenen tijdens deze eenparig cirkelvormige beweging. Het begintijdstip is gelijk aan 1,95 seconden en het eindtijdstip is gelijk aan 20,9 seconden

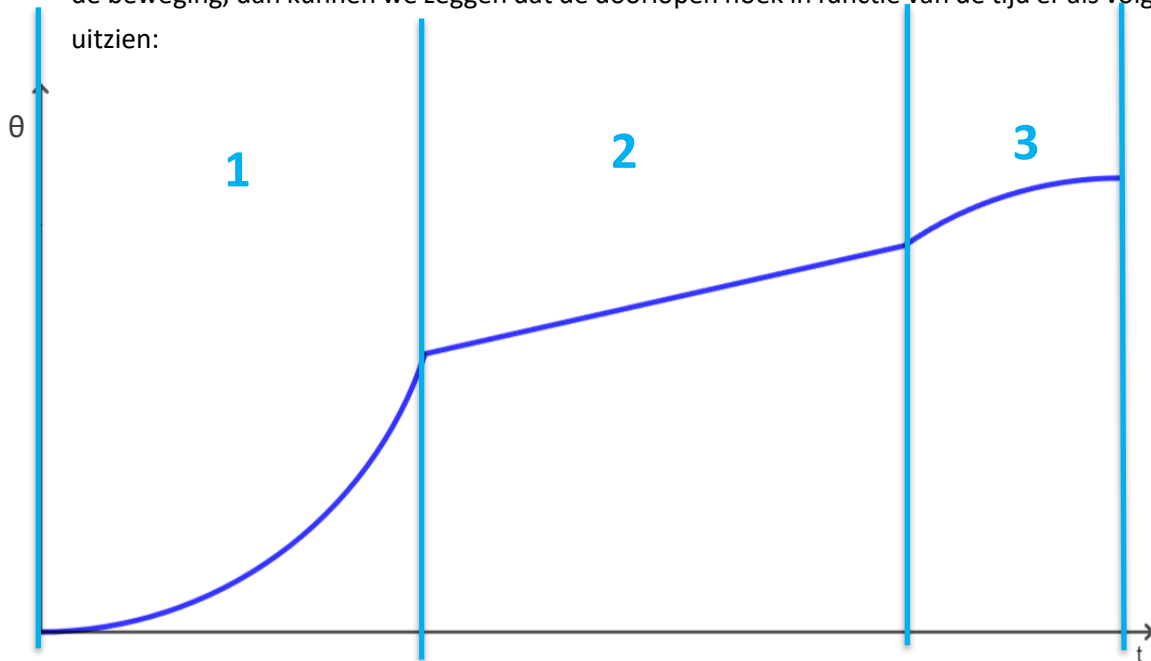
$$\theta = \theta_0 + \omega \cdot \Delta t$$

$$\theta = 0 + 5,64 \cdot (20,9 - 1,95)$$

$$106,87$$

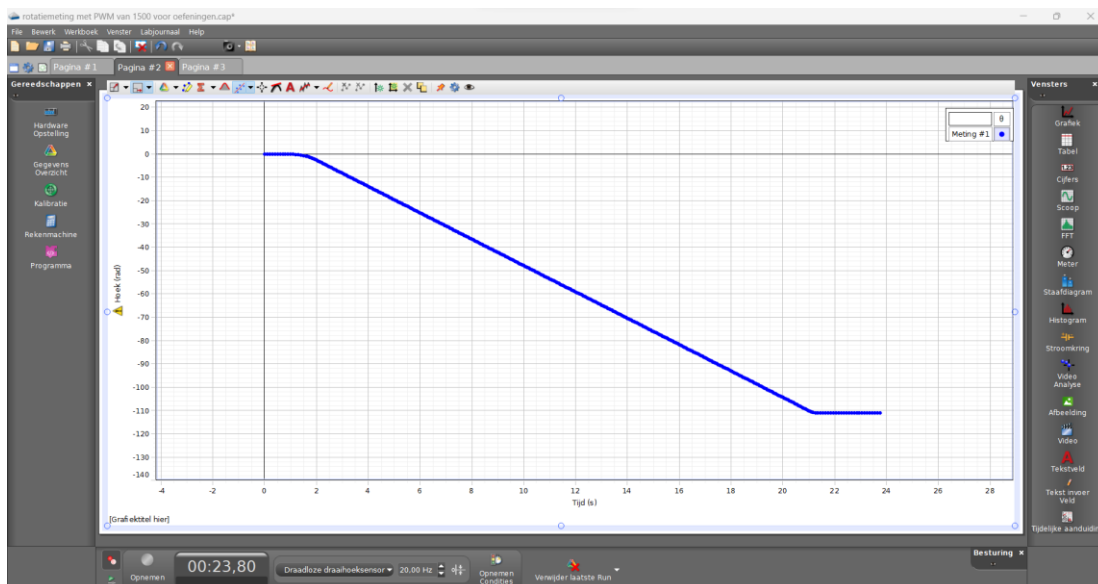
3) De doorlopen hoek i.f.v. de tijd

Als we uitgaan van een eenparig veranderlijke cirkelvormige beweging bij het begin en einde van de beweging, dan kunnen we zeggen dat de doorlopen hoek in functie van de tijd er als volgt moet uitzien:



Figuur 66 : verwachte grafiek van doorlopen hoek i.f.v. de tijd

We kunnen onze bekomen grafiek van de meting opnieuw gaan vergelijken met de theoretisch vooropgestelde grafiek. We herkennen opnieuw de 3 verschillende intervallen, waarbij telkens een andere beweging uitgevoerd wordt. Uit vorige studies van hoekversnelling en hoeksnelheid weten we al dat interval 1 geen eenparig versnelde beweging voorstelt en dat interval 3 geen eenparig vertraagde beweging voorstelt. Vandaar dat we voor de studie van doorlopen hoek ons enkel focussen op interval 2, die een eenparig cirkelvormige beweging uitvoert.



Figuur 67 : gemeten doorlopen hoek i.f.v. de tijd

We kunnen zien dat de doorlopen hoek in het midden van de grafiek een mooie lineaire functie is en dat deze dus wel eenparig verloopt.

Uit deze grafiek kunnen we nu de waarde van het begintijdstip en eindtijdstip van het lineaire gebied aflezen, alsook de waarden voor de doorlopen hoek op dat tijdstip:

begintijd: $t_1 = 1,95$ seconden $\theta_1 = 2,404$ radialen

eindtijdstip $t_2 = 20,9$ seconden $\theta_2 = 109,52$ radialen

We kunnen dus berekenen dat de doorlopen hoek in het lineaire gebied gelijk is aan:

$$109,52 \text{ rad} - 2,404 \text{ rad} = 107,16 \text{ rad}$$

Deze bekomen waarde is in overeenstemming met de waarde bekomen uit de studie van de hoeksnelheid, want $107,16 \text{ rad} \approx 106,87 \text{ rad}$.

7.3 Cycloïden

Een cycloïde is de baan afgelegd door een punt P van een rollende cirkel. Hierbij hebben we verschillende toepassingen, zo kan de cirkel langs een rechte lijnige baan rollen, maar kan hij ook rond of in een cirkelvorm rollen. In de lessen wiskunde hebben we een cycloïde gezien die op een rechte lijnige baan rolt, met onze polyp zal het wat complexer worden en hebben we te maken met een cirkel die over een cirkelvormige baan rolt.

Hieronder bespreken we de meest voorkomende cycloïden, we zullen deze op meetkundig vlak en analytisch vlak bespreken.

Eén van de doelen van de polyp was om wiskundige krommen te bekomen zoals de hieronder beschreven hypo- en epicycloïde. Deze krommen worden gevormd door het unieke punt op de tweede roterende cirkel (dit is zogezegd het karretje waarin de persoon zit). Dit doen we door specifieke stralen te gebruiken, en hier bijhorende hoeksnelheden aan toe te kennen.

7.3.1 Hypocycloïde

1) Inleiding

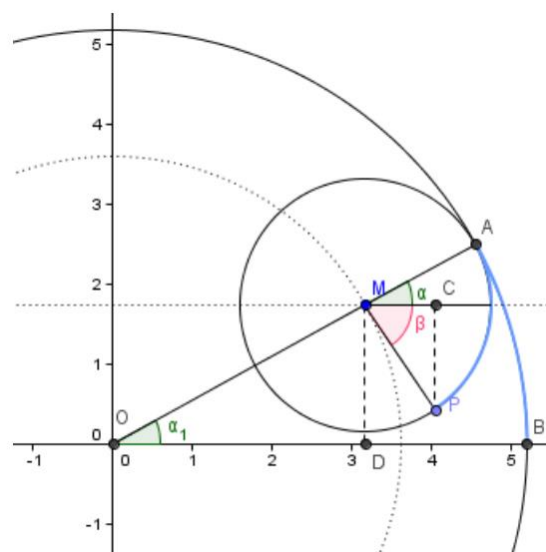
Een hypocycloïde is een cycloïde waarbij een punt op een kleine ronddraaiende cirkel in een grotere cirkel ronddraait, zoals te zien is in onderstaande afbeelding. Een hypocycloïde is dus zeer gelijkend aan een normale cycloïde, alleen draait de cirkel niet op een rechte lijnige baan, maar in een cirkelvormige baan.

2) Geschiedenis

Het woord hypocycloïde is afkomstig uit het Grieks: het woord hupo staat voor onder, kuklos voor cirkel en eidos voor vorm. Dit betekent dus: 'vorm onder cirkel'. De vorm werd ontdekt in 1599 door Galileo Galilei. De renaissance is een typische periode waarin de vooruitgang in deze krommes groot was.

3) Constructie

Bij de constructie van de hypocycloïde nemen we de oorsprong als middelpunt van de grote cirkel en R als de straal $C_1(O,R)$. Het middelpunt van de kleine cirkel is de variabele M , met straal r $C_2(M,r)$. Het beginpunt van de hypocycloïde noemen we B , het punt waar de kleine en grote cirkel elkaar raken, noemen we A en het punt dat de hypocycloïde vormt, noemen we P . Op tijdstip nul liggen de punten A en P op het punt B . De cirkelboog AB is gelijk aan cirkelboog AP omdat de cirkels niet glijden, maar rollen over elkaar. Het punt P ondergaat een dubbele rotatie, rond de oorsprong (hoek α) en rond het punt M (hoek $\alpha+\beta$).



Figuur 68 : constructie van hypocycloïde

4) Parametervergelijking opstellen

- Booglengte:

$$\widehat{AB} = \widehat{AP}$$

$$R \cdot \alpha = r \cdot (\alpha + \beta)$$

$$\frac{R}{r} \cdot \alpha = (\alpha + \beta)$$

$$k \cdot \alpha - \alpha = \beta$$

$$(k - 1) \cdot \alpha = \beta$$

- In X-richting:

$$P' = |P'O| = |M'O| + |P'M'|$$

$$\cos \alpha = \frac{|M'O|}{|MO|} \Rightarrow |M'O| = |MO| \cdot \cos \alpha = (R - r) \cdot \cos \alpha$$

$$\cos \beta = \frac{|M'P'|}{|MP|} \Rightarrow |M'P'| = |MP| \cdot \cos \beta = r \cdot \cos \beta = r \cdot \cos(k - 1)\alpha$$

$$P' = (R - r) \cos \alpha + r \cdot \cos(k - 1)\alpha$$

$$P' = (k \cdot r - r) \cos \alpha + r \cdot \cos(k - 1)\alpha$$

$$P'_x = r[(k - 1) \cos \alpha + \cos(k - 1)\alpha]$$

- In Y-richting:

$$P'' = |M''O| + |P''M''|$$

$$\sin \alpha = \frac{|M''O|}{|MO|} \Rightarrow |M''O| = |MO| \cdot \sin \alpha = (R - r) \cdot \sin \alpha$$

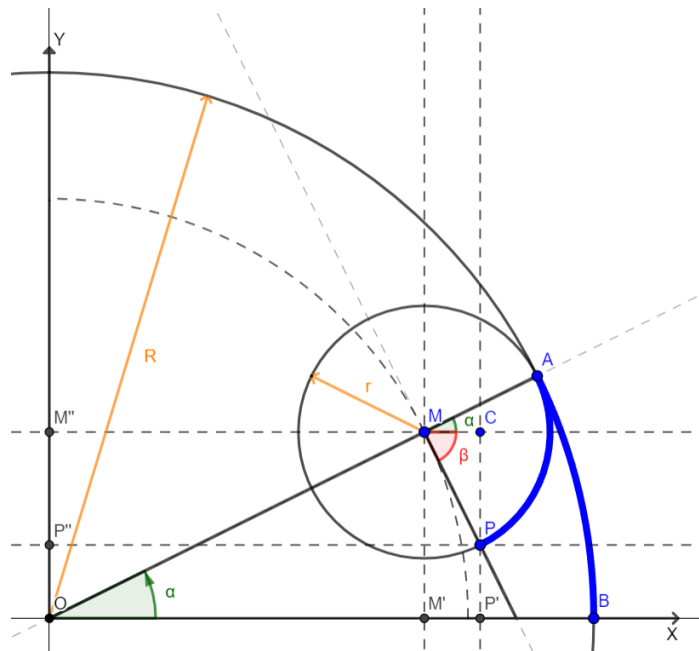
$$\sin \beta = \frac{|M''P''|}{|MP|} \Rightarrow |M''P''| = |MP| \cdot \sin \beta = r \cdot \sin \beta = r \cdot \sin(k - 1)\alpha$$

$$P'' = (R - r) \sin \alpha - r \cdot \sin(k - 1)\alpha$$

$$P''_y = r[(k - 1) \sin \alpha - \sin(k - 1)\alpha]$$

- De parametervergelijking:

$$\begin{cases} x = r[(k - 1) \cos \alpha + \cos(k - 1)\alpha] \\ y = r[(k - 1) \sin \alpha - \sin(k - 1)\alpha] \end{cases}$$

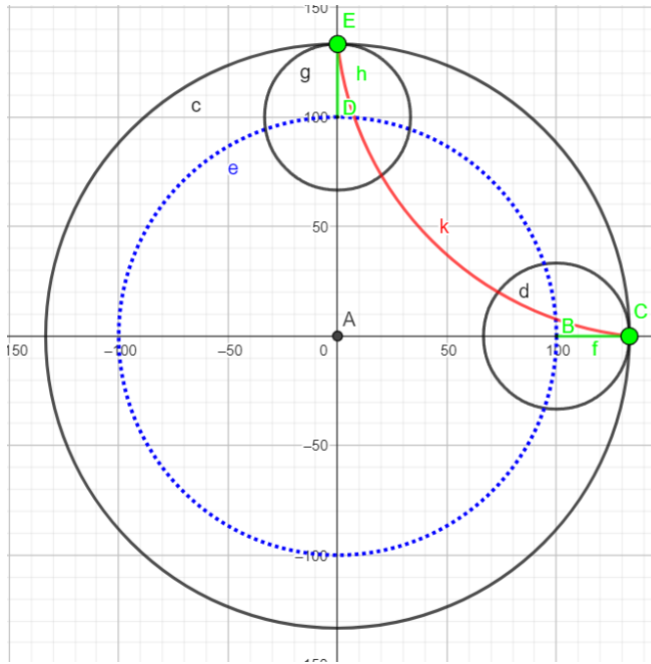


Figuur 69 : constructie hypocycloïde (bewijs PV)

5) In de praktijk

Om mooi gesloten hypocycloïden te bekommen met onze opstelling hebben we wat redeneringswerk moeten doen, dit omdat er een paar verschillen zijn tussen de theorie en de praktische realisatie.

Bepalen van de verhouding tussen de hoeksnelheid.



Figuur 70 : constructie hypocycloïde in de praktijk

cirkel d zich in de begintoestand bevinden en zal cirkel g zich in een toestand bevinden waarin de hypocycloïde al voor één vierde gevormd is.

Toestanden in het punt E

Uit bovenstaande figuur kunnen we dus besluiten dat de afgelegde hoek θ_G van de grote cirkel gelijk is aan $\frac{\pi}{2}$ radialen en de afgelegde hoek θ_k van de kleine cirkel gelijk is aan $\frac{3\pi}{2}$ radialen.

Hieruit volgt dus dat:

$$3. \theta_G = \theta_k$$

$$3. \omega_G \cdot t = \omega_k \cdot t$$

$$3. f_G \cdot 2\pi = f_k \cdot 2\pi$$

$$\frac{f_k}{f_G} = 3$$

Dit wil dus zeggen dat de kleine cirkel driemaal zo snel zal moeten draaien als de grote cirkel, ofwel dat de servomotor driemaal zo snel zal moeten draaien als de hoofdmotor.

De naastliggende figuur geeft mooi de theoretische opstelling van onze polyp weer.

- Kleine cirkel d draait in wijzerzin.
- Grote cirkel e draait in tegenwijzerzin.
- De diameter van de grote cirkel e is 200 millimeter en de diameter van de kleine cirkel g is 66,667 millimeter zo verkrijgen we een straalverhouding van één op vier.

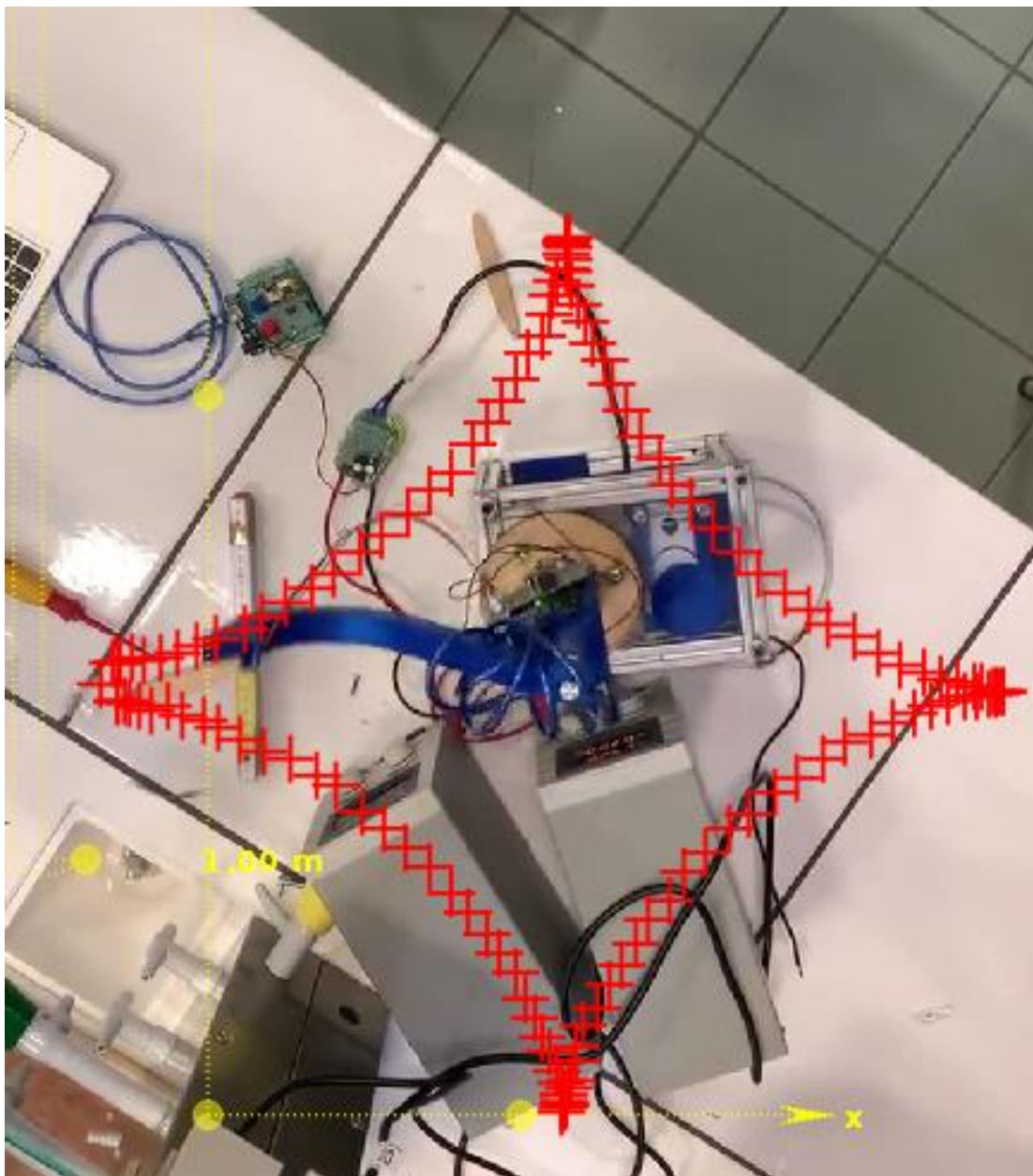
Hierbij kunnen we de blauwe cirkel e (in stippellijn) gelijkstellen aan de arm en de kleine cirkels d en g gelijkstellen aan de plaatjes op de servomotor, de kleine cirkels zijn eigenlijk dezelfde cirkels. Alleen zal de

We kunnen nu zeggen dat q gelijk is aan de verhouding van de grote straal op de kleine straal, q is dus gelijk aan 4. Hieruit volgt dus dat voor hypocycloïden $\frac{R_G}{r_k} = q = 4$

De verhouding van de kleine frequentie op de grote frequentie kunnen we dus schrijven als:

$$\frac{f_k}{f_G} = \frac{R_G}{r_k} - 1 = q - 1 = 4 - 1 = 3$$

6) Bekomen hypocycloïde



Figuur 71 : bekomen hypocycloïde

7.3.2 Epicycloïden

1) Inleiding

De epicycloïde is erg verwant aan de hypocycloïde, het verschil hier is dat het een kromme is die gevormd wordt door een punt op een cirkel, deze cirkel rolt rond een andere cirkel.

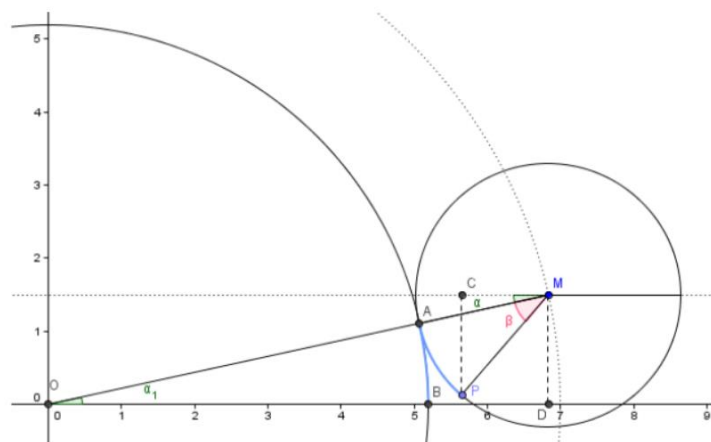
2) Geschiedenis

De naam epicycloïde is afkomstig uit het Grieks. Hierin staat epi voor bovenop, kuklos voor cirkel en eidos voor vorm. Het woord epicycloïde werd voor het eerst zo genoemd door Ole Rømer in 1674. Hiervoor werd de vorm wel al gebruikt in de planetenleer door onder andere Aristoteles en Ptolemaeus.

Later werd de kromme nog onder handen genomen door enkele bekende namen zoals: Isaac Newton, Bernoulli, Euler...

3) Constructie

Bij de constructie van de epicycloïde nemen we de oorsprong als middelpunt van de grote cirkel en R als de straal $C_1(O,R)$. Het middelpunt van de kleine cirkel is de variabele M , met straal r $C_2(M,r)$. Het beginpunt van de epicycloïde noemen we B , het punt waar de kleine en grote cirkel elkaar raken, noemen we A en het punt dat de



Figuur 72 : constructie epicycloïde

epicycloïde vormt, noemen we P . Op tijdstip nul liggen de punten A en P op het punt B . De cirkelboog AB is gelijk aan cirkelboog AP omdat de cirkels niet glijden, maar rollen over elkaar. Het punt P ondergaat een dubbele rotatie, rond de oorsprong (hoek α) en rond het punt M (hoek $\alpha+\beta$).

4) Parametervergelijking opstellen

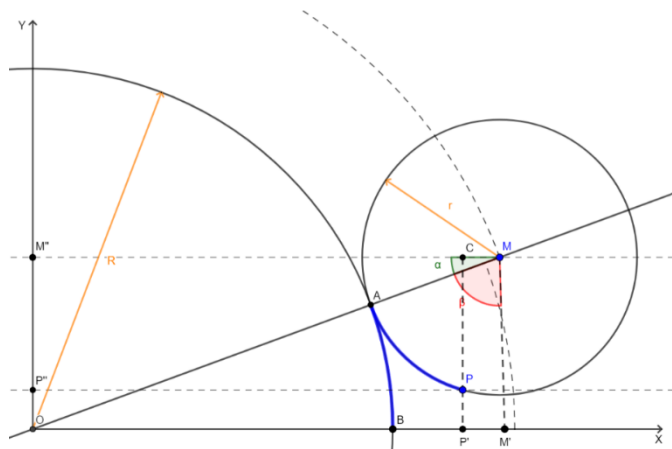
Booglengte:

$$\widehat{AB} = \widehat{AP}$$

$$R \cdot \alpha = r \cdot \beta$$

$$\frac{R}{r} \cdot \alpha = \beta$$

$$k \cdot \alpha = \beta$$



Figuur 73 : constructie epicycloïde (bewijs PV)

- In X-richting:

$$P' = |P'O| = |M'O| - |P'M'|$$

$$\cos \alpha = \frac{|M'O|}{|MO|} \Rightarrow |M'O| = |MO| \cdot \cos \alpha = (R + r) \cdot \cos \alpha$$

$$\cos(\alpha + \beta) = \frac{|M'P'|}{|MP|} \Rightarrow |M'P'| = |MP| \cdot \cos(\alpha + \beta) = r \cdot \cos(\alpha + \beta) = r \cdot \cos((k + 1)\alpha)$$

$$P' = (R + r) \cos \alpha - r \cdot \cos(k + 1)\alpha$$

$$P' = (k \cdot r + r) \cos \alpha - r \cdot \cos(k + 1)\alpha$$

$$P'_x = r[(k + 1) \cos \alpha - \cos(k + 1)\alpha]$$

- In Y-richting:

$$P'' = |M''O| - |P''M''|$$

$$\sin \alpha = \frac{|M''O|}{|MO|} \Rightarrow |M''O| = |MO| \cdot \sin \alpha = (R + r) \cdot \sin \alpha$$

$$\sin(\alpha + \beta) = \frac{|M''P''|}{|MP|} \Rightarrow |M''P''| = |MP| \cdot \sin(\alpha + \beta) = r \cdot \sin(\alpha + \beta) = r \cdot \sin((k + 1)\alpha)$$

$$P'' = (R + r) \sin \alpha - r \cdot \sin(k + 1)\alpha$$

$$P''_y = r[(k + 1) \sin \alpha - \sin(k + 1)\alpha]$$

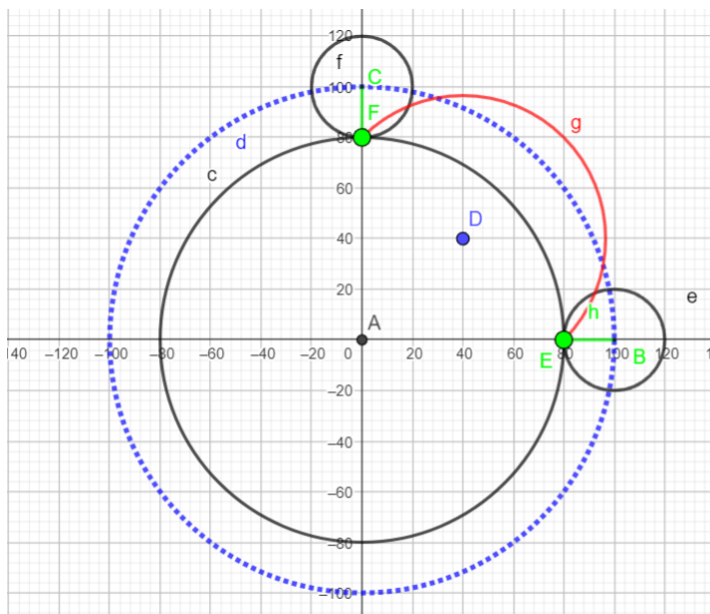
- De parametervergelijking:

$$\begin{cases} x = r[(k + 1) \cos \alpha - \cos(k + 1)\alpha] \\ y = r[(k + 1) \sin \alpha - \sin(k + 1)\alpha] \end{cases}$$

5) In de praktijk

Om een mooi gesloten epicycloïde te bekomen met onze opstelling hebben we opnieuw wat redeneringswerk moeten doen, omdat er een paar verschillen zijn in de theorie en de praktische realisatie.

1 bepalen van de verhouding tussen de hoeksnelheid.



Figuur 74 : constructie epicycloïde in de praktijk

cirkels zijn eigenlijk dezelfde cirkels, alleen zal de cirkel e zich in de begintoestand bevinden en zal cirkel f zich in een toestand bevinden waar de epicycloïde al voor één vierde gevormd is.

Toestanden in het punt F

Uit bovenstaande figuur kunnen we dus besluiten dat de afgelegde hoek θ_G van de grote cirkel gelijk is aan $\frac{\pi}{2}$ radialen en de afgelegde hoek θ_k van de kleine cirkel gelijk is aan $\frac{5\pi}{2}$ radialen.

Hieruit volgt dus dat:

$$5. \theta_G = \theta_K$$

$$5. \omega_G \cdot t_G = \omega_k \cdot t_k$$

$$5. f_G \cdot 2\pi \cdot t_G = f_k \cdot 2\pi \cdot t_k$$

$$\frac{f_k}{f_G} = 5$$

De naastliggende figuur geeft mooi de theoretische opstelling van onze polyp weer.

- Kleine cirkel e en de grote cirkel c draaien beiden in tegenwijzerzin.
- De diameter van de grote cirkel c is 160 millimeter en de diameter van de kleine cirkel e is 40 millimeter zo verkrijgen we een straalverhouding van één op vier.

Hierbij kunnen we de blauwe cirkel d (in stippellijn) gelijkstellen aan de arm en de kleine cirkels e en f gelijkstellen aan de plaatjes op de servomotor, de kleine

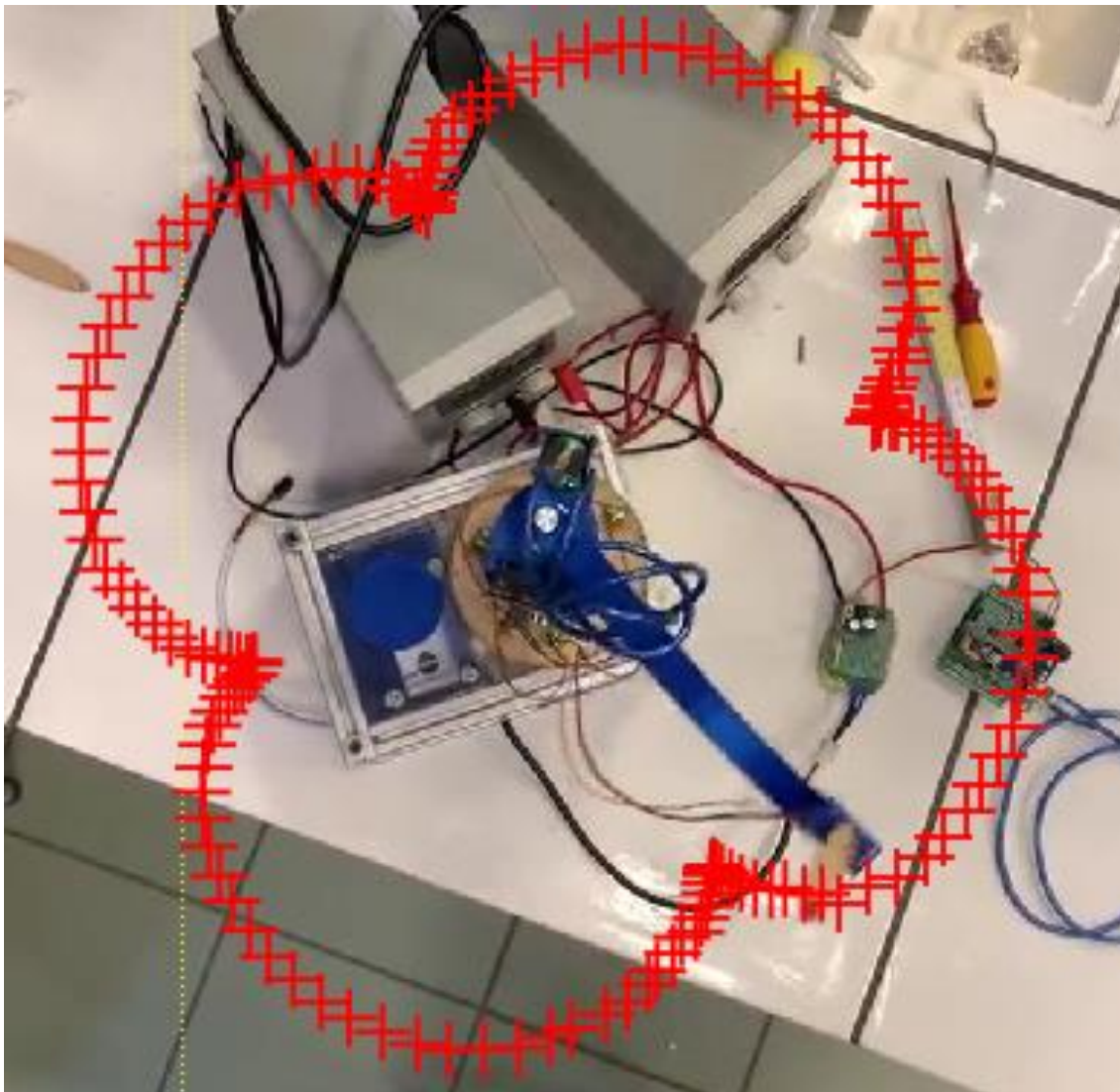
Dit wil dus zeggen dat de kleine cirkel vijfmaal zo snel zal moeten draaien als de grote cirkel, ofwel dat de servomotor vijfmaal zo snel zal moeten draaien als de hoofdmotor.

We kunnen nu zeggen dat q gelijk is aan de verhouding van de grote straal op de kleine straal, q is dus gelijk aan 4. Hieruit volgt dus dat voor epicycloïden $\frac{R_G}{r_k} = q = 4$

De verhouding van de kleine frequentie op de grote frequentie kunnen we dus schrijven als:

$$\frac{f_k}{f_G} = \frac{R_G}{r_k} + 1 = q + 1 = 4 + 1 = 5$$

6) Bekomen epicycloïde



Figuur 75 : bekomen epicycloïde

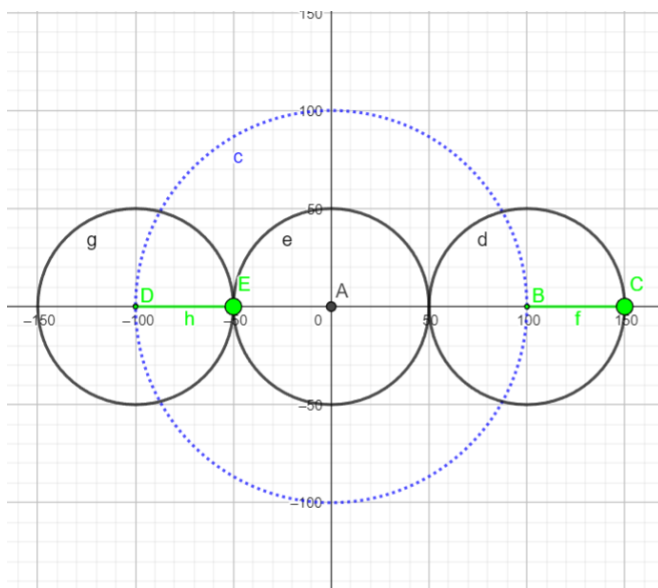
7.3.3 Cardioïden

De cardioïde, ook wel de hartlijn genoemd is een speciale vorm van een epicycloïde. Hierin zijn de stralen van de twee cirkels gelijk.

1) In de praktijk

Om cardioïden te bekomen met onze opstelling hebben we opnieuw wat redeneringswerk moeten doen, omdat er een paar verschillen zijn in de theorie en de praktische realisatie.

Bepalen van de verhouding tussen de hoeksnelheid



Figuur 76 : constructie cardioïde in de praktijk

De naastliggende figuur geeft mooi de theoretische opstelling van onze polyp weer.

- Kleine cirkel d en de grote cirkel c draaien beiden in tegenwijzerzin.
- De beide diameters van de cirkels zijn nu even groot, dit is een eigenschap van de cardioïde.

Hierbij kunnen we de blauwe cirkel c (in stippellijn) gelijkstellen aan de arm en de twee buitenste cirkels d en g kunnen we gelijkstellen aan de plaatjes op de servomotor, de buitenste cirkels zijn eigenlijk dezelfde cirkels alleen zal de cirkel

d zich in de begintoestand bevinden en zal cirkel g zich in een toestand bevinden waarin de cardioïde al voor de helft gevormd is.

Toestanden in het punt E

Uit bovenstaande figuur kunnen we dus besluiten dat de afgelegde hoek θ_G van de grote cirkel gelijk is aan π -radialen en de afgelegde hoek θ_K van de kleine cirkel gelijk is aan 2π -radialen.

Hieruit volgt dus dat:

$$2. \theta_G = \theta_K$$

$$2. \omega_G \cdot t = \omega_K \cdot t$$

$$2 \cdot f_G \cdot 2\pi = f_k \cdot 2\pi$$

$$\frac{f_k}{f_G} = 2$$

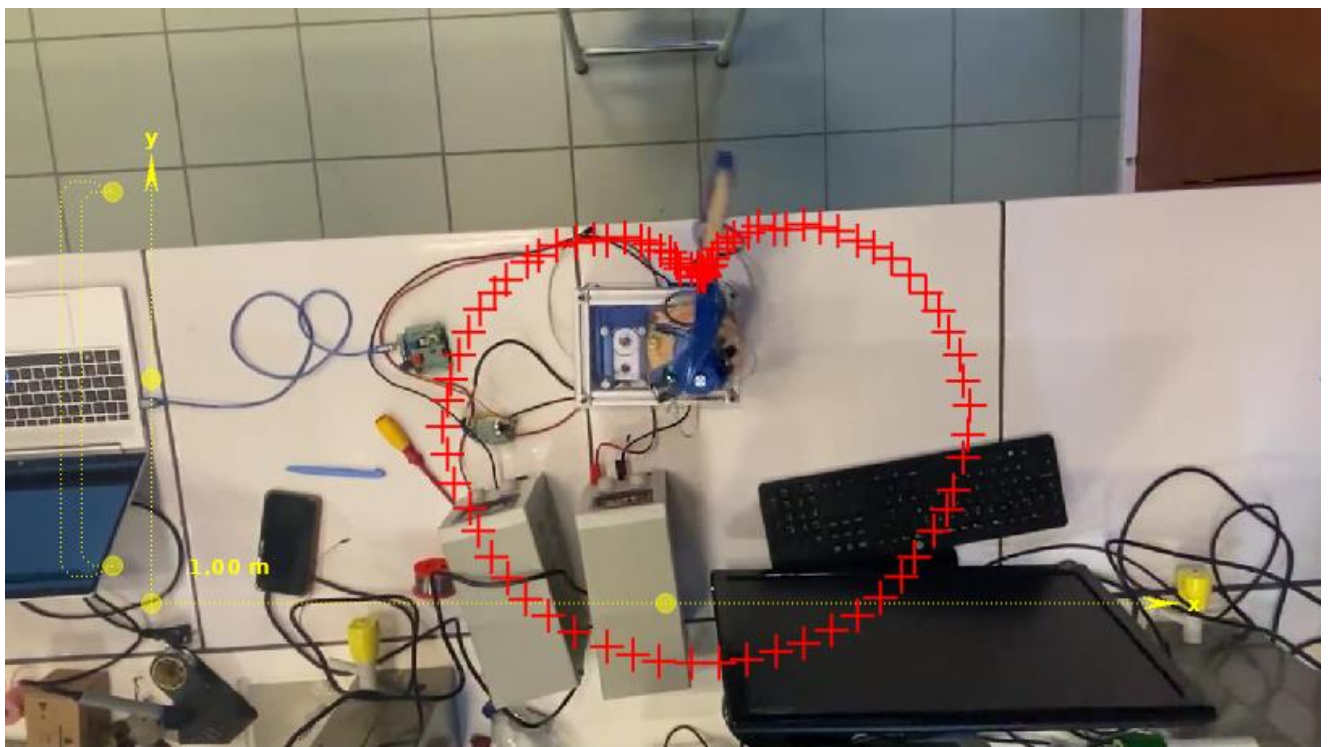
Dit wil dus zeggen dat de kleine cirkel tweemaal zo snel zal moeten draaien als de grote cirkel, ofwel dat de servomotor tweemaal zo snel zal moeten draaien als de hoofdmotor.

We kunnen nu zeggen dat q gelijk is aan de verhouding van de grote straal op de kleine straal, q is dus gelijk aan 4. Hieruit volgt dus dat voor epicycloïden $\frac{R_G}{r_k} = q = 1$.

De verhouding van de kleine frequentie op de grote frequentie kunnen we dus schrijven als:

$$\frac{f_k}{f_G} = \frac{R_G}{r_k} + 1 = q + 1 = 1 + 1 = 2$$

2) De bekomen cardioïde



Figuur 77 : bekomen cardioïde

8. ICT-studie

8.1 Website

Voor de 3 GIP's heeft Wout Crevits ook nog een site geschreven om wat meer info te verschaffen over de verschillende leerlingen van 6IW en over de verschillende bedrijfsbezoeken. Deze site is gebaseerd op een template die hij online vond, gemaakt door HTMLup.net.

De site is zoals de meeste met HTML, CSS en Javascript geschreven waarbij html voor de tekst dient, css dient voor de layout en "looks" terwijl javascript zorgt voor de functie en werking van de site.

Ziehier wat foto's van de site en de code.

```
        <a href="#banner_OnzeGIP" class="button circled scrolly">Start</a>
    </footer>
</div>

<!-- Nav -->
<nav id="nav">
    <ul>
        <li><a href="index.html">Home</a></li>
        <li>
            <p>Onze GIP</p>
            <ul>
                <li><a href="Aqualab.html">Aqualab</a></li>
                <li><a href="Airco.html">Airco</a></li>
                <li><a href="Polyp.html">Polyp</a></li>
            </ul>
        </li>
        <li><a href="Activiteiten.html">Activiteiten</a></li>
        <li><a href="archieff_main.html">Archieff</a></li>
    </ul>
</nav>

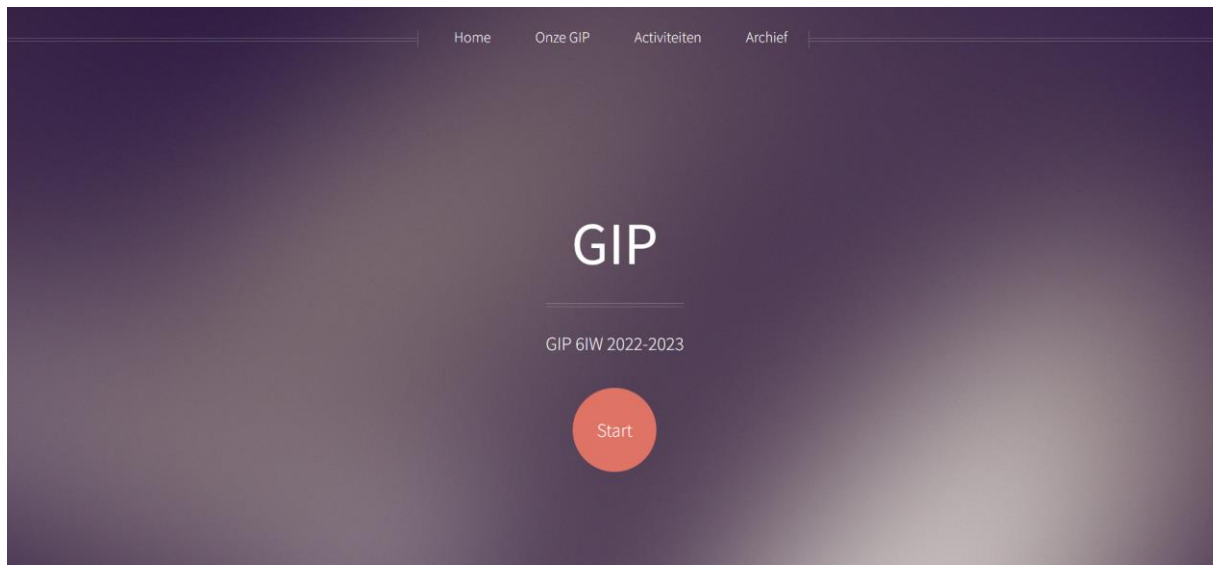
</div>

<!-- Banner -->
<section id="banner_OnzeGIP">
    <header>
        <h2>Onze GIP</h2>
        <p>Voor onze GIP maken we 3 verschillende projecten, namelijk het Aqualab, de Airco en de Polyp.</p>
        <p>Deze projecten bevatten bepaalde stukken leerstof, die we te zien krijgen in Industriële Wetenschappen.</p>
    </header>
</section>

<!-- Main -->
<div class="wrapper style2">

    <article id="main" class="container special">
```

Figuur 78 : code programma van de site



Figuur 79 : site

Eerste presentatie

Vond plaats op 12/09/2022

Dit was de eerste keer dat de 3 groepen hun GIP voorstelden aan de jury. We maakten aan de hand van een powerpoint duidelijk wat ze wilden bereiken met hun GIP en wat het eerste idee was om dit te realiseren. Hierop gaf de jury de nodig kritiek om de leerlingen verder opweg te helpen.

[Klik hier voor foto's.](#)

Eerste projectweek



Figuur 80 : site

De link naar de site is: GIP6IW.be. Je kan op de site ook de powerpoints van de presentatie vinden.

9. Evaluatie – besluit

Tot slot bekijken we het totaalplaatje. We kregen in het begin van het schooljaar het voorstel om een polyp te maken. We zijn direct beginnen brainstormen over wat de polyp is en wat we allemaal wilden realiseren en aantonen. Al snel besloten we dat we voor 2 meetopstellingen zouden gaan. We wilden een meetstelling om de krachten te meten en een meting om de cycloïden te bepalen. We gingen normaal een pretpark bezoeken als bedrijfsbezoek om wat info te krijgen over deze attractie, maar dat kon door omstandigheden jammer genoeg niet doorgaan.

Tijdens het schooljaar vlogen we er direct in. Al snel hadden we een paar goeie schetsen gemaakt en onze ideeën verder uitgewerkt, waardoor we konden beginnen tekenen in solid edge. Door een goeie basisconstructie in solid edge konden we snel zien hoe we de constructies voor de metingen konden bevestigen. Al snel zagen we een probleem bij de draaibeweging in, namelijk de voeding voor de servomotoren. Deze moeten op de draaiende as komen. Na overleg met dhr. Geeraert hebben we besloten te werken via sleepcontacten. We waren zeer enthousiast over dit idee omdat we daar nog nooit van gehoord hadden en dat graag eens wilden proberen.

Na het ontwerpen van de connectie van de as en de sleepcontacten konden we beginnen denken over hoe de meetopstellingen konden bevestigd worden. Eenmaal deze getekend waren, kon de constructie snel gemaakt worden doordat de stukken allemaal gemaakt konden worden via de 3D-printer en lasercutter. Eenmaal de constructie vorm kreeg, konden we beginnen met onze metingen uit te voeren.

Dit klinkt allemaal wel leuk, maar gemakkelijk is een GIP niet. We kwamen het ene na het andere probleem tegen dat we dan weer moesten oplossen. Soms duurde het enkele weken vooraleer we een oplossing vonden.

Uiteindelijk hebben we mooie metingen kunnen bekomen en zijn we zeer tevreden over het eindresultaat waar we een heel jaar hard voor gewerkt hebben. We hopen dat de jaren na ons dankzij ons project de leerstof beter kunnen verwerken.