

Academiejaar  
2022 - 2023

# Flight controller

## UAntwerp Solar Boat Team

**Devlieghere Mattijs**

Bachelorproef

**Bachelor of Science in de industriële wetenschappen: elektromechanica**

Promotor

**Van Thillo, Jitse, Doctoraatsstudent EMIB Universiteit Antwerpen**



**Universiteit Antwerpen**  
| Faculteit Toegepaste  
Ingenieurswetenschappen

## Disclaimer Bachelorproef

Dit document is een examendocument dat niet werd gecorrigeerd voor eventueel vastgestelde fouten. Zonder voorafgaande schriftelijke toestemming van zowel de promotor(en) als de auteur(s) is overnemen, kopiëren, gebruiken of realiseren van deze uitgave of gedeelten ervan verboden. Voor aanvragen tot of informatie i.v.m. het overnemen en/of gebruik en/of realisatie van gedeelten uit deze publicatie, wendt u tot de universiteit waaraan de auteur is ingeschreven.

Voorafgaande schriftelijke toestemming van de promotor(en) is eveneens vereist voor het aanwenden van de in dit afstudeerwerk beschreven (originele) methoden, producten, schakelingen en programma's voor industrieel of commercieel nut en voor de inzending van deze publicatie ter deelname aan wetenschappelijke prijzen of wedstrijden.

Dit document is in overeenstemming met het bachelorproefreglement en de gedragscode en nagekeken door promotor en begeleider.

# Voorwoord

Graag wil ik dit voorwoord beginnen met mijn promotor, Jitse Van Thillo, te bedanken. Hij wist steeds raad op de momenten dat ik vastzat en was altijd beschikbaar. Verder konden wij samen nadenken over complexe zaken om de beste uitkomst te garanderen. Zijn expertise als vroegere teamleider van het UAntwerp Solar Boat Team droeg steeds bij tot interessante nieuwe invalshoeken.

Hoewel ik nog maar één jaar in het UAntwerp Solar Boat team zit, heb ik al enorm veel bijgeleerd. Zo leerde ik onderhandelen met sponsors, regeltechnische begrippen, carbon maken, plamuren en zo veel meer. Het warme welkom van alle teamleden zorgde ervoor dat ik meteen mijn plaats vond. Ik wil dan ook Seppe Van den Broeck bedanken om mij goed te begeleiden en altijd hulp te bieden waar nodig.

Graag bedank ik ook nog al mijn vrienden en familie die steeds klaarstonden om te luisteren.

Volgend jaar word ik deel van de teamleiding. Bedankt aan het team voor dit vertrouwen.

# Abstract

Het UAAntwerp Solar Boat Team is een team van studenten industriële wetenschappen aan de Universiteit Antwerpen, dat samen een solar boot bouwt en verder ontwikkelt, met oog op duurzaamheid en energie-efficiëntie.

Dit team implementeert hydrofoils (draagvleugels) om de vaart efficiënter te laten verlopen, door tijdens de vaart de boot uit het water te tillen.

Deze bachelorproef onderzoekt het elektronisch regelsysteem voor een met motor gestuurd hydrofoilsysteem. Er werd onderzoek uitgevoerd naar het inlezen van sensordata, het filteren van deze data, het uitbouwen van het regelsysteem (zowel de software als de hardware), de bepaling van de parameters en mogelijke veiligheidsmaatregelen. Het inlezen en filteren van de sensordata werd eveneens uitgebouwd.

Er wordt, als inputs van de regelaar, gebruik gemaakt van twee ultrasone sensoren en één gyroscoop. De sensordata wordt gefilterd aan de hand van een low-pass filter. Deze filter werd uitgebouwd op een Arduino Due en Arduino Uno. Deze Arduinos bleken beide niet geschikt.

Het regelsysteem is uitgebouwd door gebruik te maken van PID-regelaars in combinatie met een co-simulatie door middel van Siemens NX Motion en Simulink. Het bepalen van de parameters van deze regelaars werd uitgevoerd door middel van een iteratie en PSO-algoritme. Uit de resultaten kan worden afgeleid dat de co-simulatie instabiel en omslachtig is.

Hoewel de parameters geoptimaliseerd zijn, blijken de regelaars te traag en moeten andere typen regelaars in volgend onderzoek onderzocht worden.

Ten slotte werd ook de veiligheid tijdens het foilen in acht genomen. Hierbij worden enkele voorschriften, zoals een noodstopprotocol, beschreven waar verder onderzoek rekening mee moet houden.



# Symbolen en afkortingen

|                  |   |
|------------------|---|
| CAN              | Controller Area Network   |
| DOF              | Degrees Of Freedom, aantal vrijheidsgraden  |
| F                | Transfer functie van een low-pass filter  |
| $h_{corr}$       | Gecorrigeerde hoogte meting   |
| $h_{gem,corr}$   | Gemiddelde van de gecorrigeerde hoogte meting                                     |
| $h_{ref}$        | Lopende gemiddelde over vier golven van $h_{gem,corr}$                            |
| $h_{reg}$        | $h_{SP}$ ten opzichte van $h_{ref}$   |
| $h_{sensor}$     | Hoogte gemeten door een ultrasone sensor  |
| $h_{SP}$         | Setpoint hoogte   |
| LQR              | Linear-Quadratic Regulator  |
| MIMO             | Multiple Inputs Multiple Outputs, systeem met meerdere inputs en meerdere outputs |
| MPC              | Model Predictive Control  |
| NaN              | Not a Number  |
| PID              | Proportionele Integrator Differentiële (regelaar)                                 |
| PSO              | Particle Swarm Optimization   |
| SISO             | Single Input Single Output, systeem met één input en één output                   |
| SP               | Setpoint, gewenste waarde   |
| $\omega_0$       | Drempelfrequentie   |
| $\theta_{pitch}$ | Pitch hoek  |
| $\theta_{roll}$  | Roll hoek   |

# Inhoudsopgave

|  |            |
|--|------------|
| <b>Voorwoord</b>                               | <b>i</b>   |
| <b>Abstract</b>                                | <b>ii</b>  |
| <b>Symbolen en afkortingen</b>                 | <b>iii</b> |
| <b>1. Inleiding</b>                            | <b>1</b>   |
| 1.1. UAntwerp Solar Boat Team . . . . .        | 1          |
| 1.2. Hydrofoiling . . . . .                    | 1          |
| 1.3. Regelaar . . . . .                        | 3          |
| <b>2. Filtering</b>                            | <b>4</b>   |
| 2.1. Inleiding . . . . .                       | 4          |
| 2.2. Ultrasonische sensoren . . . . .          | 4          |
| 2.2.1. Ultrasonische metingen - Boot . . . . . | 4          |
| 2.2.2. Geen referentie op het water . . . . .  | 5          |
| 2.2.3. X van Y filter . . . . .                | 5          |
| 2.3. Gyroscop . . . . .                        | 6          |
| 2.3.1. Problemen . . . . .                     | 6          |
| 2.4. Platform . . . . .                        | 6          |
| 2.5. Filtering . . . . .                       | 7          |
| 2.6. Testen . . . . .                          | 9          |
| 2.7. Bedrading . . . . .                       | 9          |
| <b>3. Regeling</b>                             | <b>10</b>  |
| 3.1. Inleiding . . . . .                       | 10         |
| 3.2. Model . . . . .                           | 10         |
| 3.2.1. Siemens NX Motion - Simulink . . . . .  | 11         |
| 3.3. Regelaar . . . . .                        | 11         |
| 3.3.1. Open lus - Gesloten lus . . . . .       | 11         |
| 3.3.2. Types regelaar . . . . .                | 11         |
| 3.3.3. Ontwerp regelaar . . . . .              | 12         |
| 3.4. Parameters bepalen . . . . .              | 12         |
| 3.4.1. Iteratie . . . . .                      | 13         |
| 3.4.2. Particle Swarm Optimization . . . . .   | 14         |
| 3.4.3. Resultaten . . . . .                    | 15         |
| 3.5. Veiligheid . . . . .                      | 16         |
| <b>4. Conclusie</b>                            | <b>17</b>  |
| 4.1. Besluit . . . . .                         | 17         |
| 4.2. Toekomst . . . . .                        | 17         |
| <b>Bibliografie</b>                            | <b>19</b>  |
| <b>A. Appendix</b>                             | <b>20</b>  |

# 1. Inleiding

## 1.1. UAntwerp Solar Boat Team

Het UAntwerp Solar Boat Team werkt elk academiejaar aan de verbeteringen van een boot die aangedreven wordt door zonne-energie. Dit team racet elk jaar met deze boot in internationale competities. Het team van het academiejaar 2022-2023, bestaande uit 21 studenten toegepaste ingenieurswetenschappen elektromechanica (zie figuur: 1.1), zal dit jaar deelnemen aan de 'Monaco Solar & Energy Boat Challenge'. Het team streeft naar een steeds duurzamere en efficiëntere toekomst van de boot.



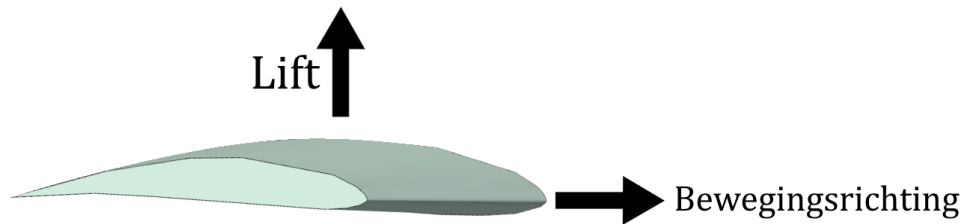
Figuur 1.1.: Teamfoto UAntwerp Solar Boat Team 2022-2023

De leden in dit team verwerven een op duurzaamheid gerichte attitude. Bovendien kan het onderzoek door de teamleden een impact hebben op nieuwe duurzame technologieën. Zo heeft bijvoorbeeld een efficiënt batterijpakket onder andere toepassingen in de auto-industrie [1]. Het hoofddoel van de verbeteringen die het team dit jaar wil realiseren is: hydrofoilen in Monaco.

## 1.2. Hydrofoiling

Hydrofoilen is het effect dat men krijgt wanneer men foils (draagvleugels) onder de boot monteert die een lift genereren tijdens het varen, waardoor de boot boven het water wordt geheven. Draagvleugels zijn zeer gelijkaardig aan vliegtuigvleugels, waarbij draagvleugels lift creëren in water zoals vliegtuigvleugels lift creëren in de lucht.

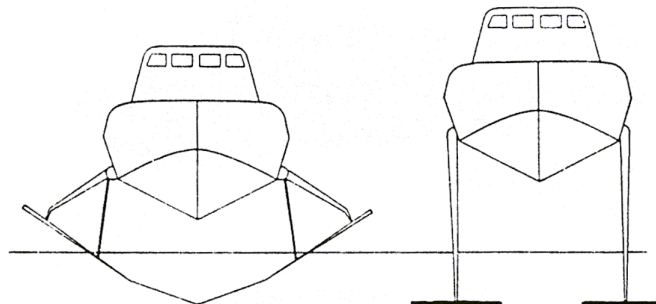
Figuur 1.2 geeft een hydrofoil weer. Wanneer een hydrofoil op zijn foilsnelheid beweegt, creëert deze een liftkracht loodrecht op de bewegingsrichting, naar boven [2].



Figuur 1.2.: Hydrofoil met de bewegingsrichting en liftrichting

Wanneer een boot foilt, maakt deze, behalve de draagvleugels en struts, geen contact meer met het water en kan de wrijvingsweerstand tot 25,7% dalen [3]. In een competitie waar men nauw moet letten op energieverbruik is dit enorm impactvol.

Er zijn twee types draagvleugels: enerzijds het oppervlakte doorborende type en anderzijds de volledig ondergedompelde variant (zie figuur 1.3). Het oppervlakte doorborende type bestaat uit één grote foil in een v-vorm. Hierbij steken, tijdens het foilen, delen van de foil uit het water. Als de boot te veel zakt, is er een groter deel van de foils onder water, waardoor de lift groter wordt en de boot opnieuw stijgt. Een gelijkaardig effect doet zich voor wanneer de boot te veel stijgt of naar één richting rolt. Het oppervlakte doorborende type is dus een zelfstabiliserend systeem, maar beweegt sterker tijdens het varen (zowel op en neer, als een grotere rolbeweging) dan het volledig ondergedompelde type. De foils van het volledig ondergedompelde type bevinden zich altijd volledig onder het wateroppervlak. De staven die de foils met de boot verbinden (struts) dragen niet bij tot de totale lift. Het volledig ondergedompelde type is inherent instabiel (een regelsysteem is nodig), maar deze is wel kleiner (en dus lichter) en laat de boot, mits een goede regelaar, minder bewegen (zowel op en neer, als de rolbeweging) <sup>1</sup> [5]. Het team koos in het verleden voor de volledig ondergedompelde variant te bouwen wegens de juist vernoemde voordelen.



Figuur 1.3.: Links: Oppervlakte doorborende. Rechts: Volledig ondergedompeld. [5]

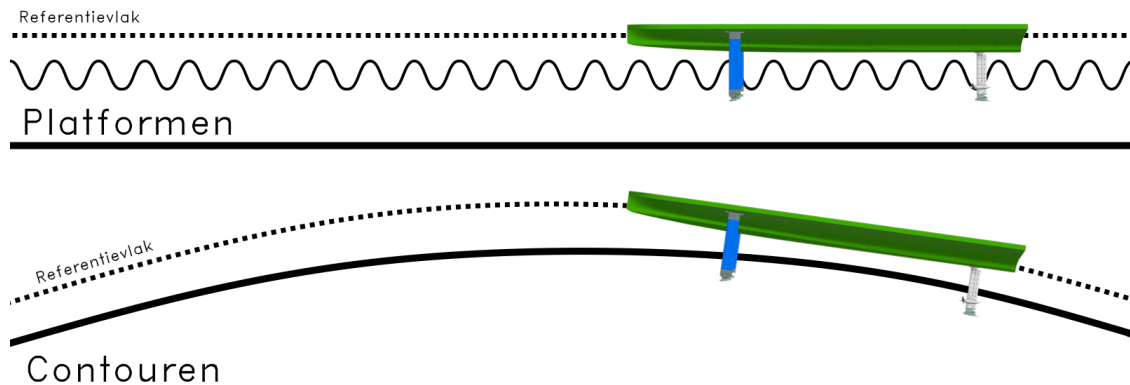
Er zijn twee manieren om te foilen; platformen en contouren. Bij het platformen snijdt de boot door de golven waardoor deze op een vaste hoogte ten opzichte van een referentievlak blijft. Bij het contouren volgt de boot de golven, dit wordt toegepast wanneer de golven groter zijn dan de boot (zowel in de hoogte als in de lengte) [5].

De golfhoogte in Monaco in juli, wanneer de Monaco Solar & Energy Boat Challenge plaatsvindt, wordt minder dan twee procent van de tijd groter dan één meter [6] (wat kleiner is dan de verticale grootte van de boot inclusief de foils). Daarom werd er gekozen om de boot te

<sup>1</sup>Het oppervlakte doorborende type gedraagt zich sterk als een omgekeerde slinger die ook inherent instabiel is [4].

### 1.3. REGELAAR

laten platformen. Als de golven toch groter blijken, zal de regeling de boot automatisch laten contouren.



Figuur 1.4.: Boven: Platformen. Onder: Contouren. [5]

### 1.3. Regelaar

Het hydrofoilsysteem bestaat uit twee deelsystemen: het mechanische deel en het regelsysteem. Seppe Van den Broeck, een masterstudent, bouwt het mechanische deel uit. Tijdens het maken van deze bachelorproef was dit systeem nog niet uitgebouwd. Deze bachelorscriptie betreft het regelsysteem. Dit regelsysteem wordt ‘de flight controller’ genoemd aangezien de boot, door het foilen, boven het water ‘vliegt’.

De flight controller bestaat op zijn beurt ook uit twee delen; de sensoren inclusief de ruisonderdrukking en de regelaar.

De input van de sensoren wordt ingelezen in een microcontroller en vervolgens wordt er op deze data een low-pass filter toegepast. Vervolgens wordt deze sensordata doorgestuurd naar de regelaar, die de juiste signalen naar de motors stuurt om de foils in de juiste positie te manoeuvreren. Hierdoor zal de boot zich naar de gewenste positie bewegen.

Wegens een tekort aan financiële middelen kan het mechanische deel dit jaar niet aangestuurd worden door motoren. Dit heeft tot gevolg dat deze bachelorproef voornamelijk theoretisch is en nog niet wordt geïmplementeerd in de boot.

Het doel van deze bachelorproef is het bouwen van een kader voor een regelaar die zorgt voor een stabiele ‘vlucht’ van de UAntwerp Solar Boat, zodat deze in het academiejaar 2023-2024 verder onderzocht en geïmplementeerd kan worden.

## 2. Filtering

### 2.1. Inleiding

Een regelaar heeft, zoals reeds vermeld, sensordata nodig om een goede regeling te bekomen. Wanneer een foilende boot geregeld wordt, worden er twee types sensoren gebruikt; ultrasone sensoren en een gyroscoop [5].

De ultrasone sensoren meten de hoogte van de boot ten opzichte van het wateroppervlak. De gyroscoop meet de oriëntatie van de boot ten opzichte van een vast referentiekader dat tijdens een kalibratie bepaald werd. In een ideale regeling wordt de snelheid en de versnelling van de boot ook gemeten en dienen deze mee als input van de regelaar. Zonder deze data kan de boot, hoewel instabieler, toch foilen. Om dit onderzoek, en de werklust, binnen het kader van een bachelorproef te houden, werd er gekozen om de laatste twee metingen nog niet te gebruiken.

### 2.2. Ultrasone sensoren

Er zijn twee complicaties wanneer men de absolute hoogte van de boot wilt meten; een boot vaart zelden evenwijdig ten opzichte van het wateroppervlak en het wateroppervlak is geen vaste referentie. Om deze problemen op te lossen maken we o.a. gebruik van twee ultrasone sensoren (zie appendix figuur A.2).

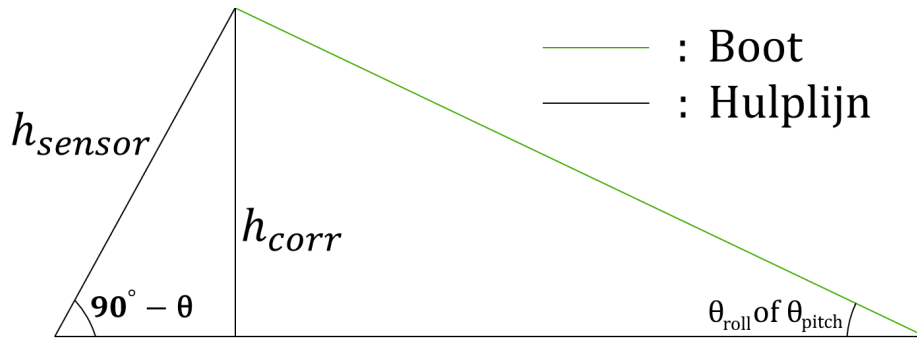
Dit wordt in paragrafen 2.2.1 en 2.2.2 verder uitgediept.

#### 2.2.1. Ultrasone metingen - Boot

Omdat de boot quasi nooit horizontaal ten opzichte van het wateroppervlak is, zullen ook de sensoren vaak schuin hangen. Hierdoor meten zij een langere afstand dan werkelijk toepasselijk is. Om dit te corrigeren wordt formule 2.1 toegepast. De oorsprong van het assenstelsel dat gebruikt wordt om de verschillende hoeken te bepalen, bevindt zich in het rotatiecentrum van de boot (zie afbeelding 2.2). Op dit punt wordt dan ook de gyroscoop bevestigd.

$$h_{corr} = h_{sensor} \cdot \sin\left(\frac{\pi}{2} - \theta_{roll}\right) \cdot \sin\left(\frac{\pi}{2} - \theta_{pitch}\right) \quad (2.1)$$

Deze formule werd opgesteld aan de hand van stellingen uit de driehoeksmeetkunde. Een verklaring van de formule valt af te leiden uit onderstaande figuur 2.1:



Figuur 2.1.: Gecorrigeerde hoogtemeting (één dimensie)

Omdat de hoogte van het water ten opzichte van de romp op verschillende punten kan variëren, nemen we het gemiddelde van de data van de twee sensoren (zie formule 2.2).

$$h_{gem,corr} = \frac{h_{corr,sensor1} + h_{corr,sensor2}}{2} \quad (2.2)$$

### 2.2.2. Geen referentie op het water

De boot moet zich ten alle tijde op één bepaalde hoogte ten opzichte van een referentievlak bevinden aangezien de boot moet platformen zoals beschreven in paragraaf 1.2. Omdat het water golft, geven de hoogtemetingen ( $h_{gem,corr}$ ) alleen de hoogte van de boot ten opzichte van het wateroppervlak op dat moment. Het kan dus voorvallen dat de boot zich werkelijk hoog bevindt, maar ten opzichte van een hoge golf, heel dichtbij is. Als de boot dan, door de regelaar, nog meer omhoog wordt gestuurd, kan het voorkomen dat de draagvleugels uit het wateroppervlak komen. Hierdoor zou de boot neervallen, dit is schadelijk voor de boot en is niet energie-efficiënt.

Om wel een vast referentievlak te bekommen, wordt het lopende gemiddelde van de  $h_{gem,corr}$  over  $X$  maal<sup>1</sup> de lengte van de boot genomen. Dit verzekert dat de microcontroller door de kleine golven, waarover er geplatformd kan worden, een referentievlak neemt, maar dat de boot wel contourt over de golven die langer zijn dan de boot. De lengte van dit interval verandert naargelang de periode van de golven, de snelheid van de boot en de vaarrichting. Volgend onderzoek moet nog uitwijzen hoe impactvol de waarde van  $X$  is en of deze al dan niet vast gekozen kan worden. De waarde die naar de regelaar wordt gezonden, is het verschil tussen de afstand tot dit referentievlak en de gewenste waarde (setpoint, SP) (zie formule 2.3).

$$h_{reg} = h_{SP} - h_{ref} \quad (2.3)$$

### 2.2.3. X van Y filter

De gebruikte ultrasonische sensoren zijn van het type: 'ToughSonic 12'. Deze sensoren hebben al een eerste filter ingebouwd, de zogenoemde 'X van Y filter'. Deze negeert alle data (en zendt geen data uit) waarvan op zijn minst  $X$  metingen van de vorige  $Y$  metingen niet binnen de 6.25% van de laatst gemeten waarde zit [7]. Voor  $X$  werd er één gekozen en voor  $Y$  werd drie gekozen. Dit werd reeds gekozen door oud-teamleden. Er moet nog meer onderzoek gebeuren of dit de optimale parameters zijn. Door deze filter hebben meetfouten en kleine druppels water, die opspringen, minder impact op de regeling.

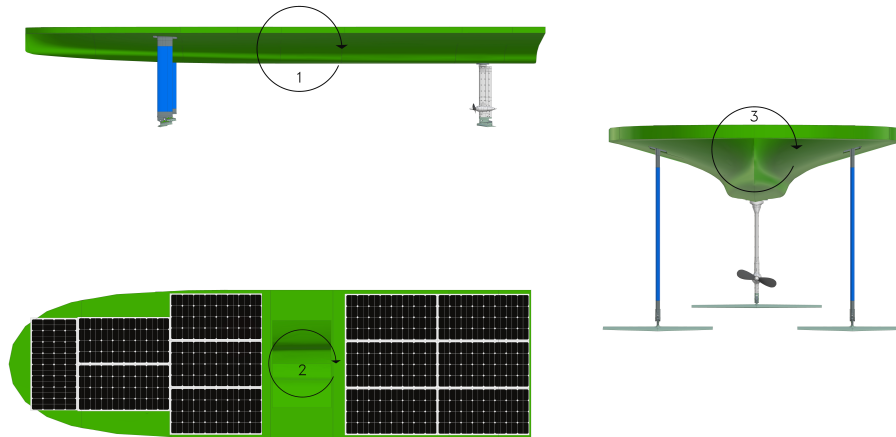
<sup>1</sup>De waarde van  $X$  moet in volgend onderzoek nog bepaald worden.

## 2.3. Gyroscop

Om te weten hoe de boot georiënteerd is in het water, (zie figuur: 2.2), kan men de hoogtesensoren gebruiken (dit is niet accuraat wegens de complicaties beschreven in paragraaf 2.2) of kan men een gyroscop gebruiken.

Daarnaast wordt de oriëntatie van de boot gebruikt om  $h_{corr}$  te berekenen (zie formule: 2.1). Een gyroscop is dus een essentiële sensor in een flight controller.

Onderstaande figuur 2.2 geeft de benamingen voor de verschillende rotatieassen en hun oorsprong in de boot weer (pitch, yaw en roll).



Figuur 2.2.: 1:Pitch, 2:Yaw en 3:Roll

### 2.3.1. Problemen

Een (6-DOF) gyroscopsensor geeft als output de versnellingen rond elke rotatieas en in de richting van elke as, waarbij de oorsprong van dit assenstelsel in het rotatiecentrum van de boot zit (zoals beschreven in paragraaf 2.2.1). Om hieruit de absolute oriëntatie te berekenen, moet men het signaal twee maal integreren over de tijd. Meetfouten in signalen wanneer de gyroscop bijna niet beweegt en hoge frequentie ruis worden ook geïntegreerd. Hierdoor worden de meetfouten steeds groter. Deze data moet daarom goed gefilterd worden. Het fenomeen waar de gyroscop niet meer de absolute oriëntatie, maar een verschoven oriëntatie uitzendt, noemt men 'gyro drift' [8]. De gebruikte gyroscop (BNO055 Adafruit 9-DOF Absolute Orientation Sensor) heeft deze problemen niet. Deze maakt gebruik van een ARM processor op de chip om alle berekeningen zelf uit te voeren. Hierdoor wordt de microcontroller minder belast en bekomen we meer accurate data [9].

## 2.4. Platform

We gebruiken het Arduino platform om de data te filteren en om de regeling te bepalen. We kozen voor het Arduino platform wegens de uitgebreide libraries en hun breed inzetvermogen. Bovendien is er veel kennis van Arduinos in het team waardoor eventuele problemen steeds besproken kunnen worden.

Binnen het Arduino platform is er een groot aanbod aan microcontrollers. Het team heeft drie typen op voorraad, namelijk: Arduino Uno, Arduino Mega en Arduino Due. Tabel 2.1 vergelijkt enkele eigenschappen van deze drie microcontrollers.



| Type Arduino | Opslaggeheugen [kB] | SRAM [kB] | # processor bits |
|--------------|---------------------|-----------|------------------|
| Arduino Uno  | 32                  | 2         | 8                |
| Arduino Mega | 256                 | 8         | 8                |
| Arduino Due  | 512                 | 96        | 32               |

Tabel 2.1.: Vergelijking van het geheugen en het aantal processor bits van de Arduino Uno, Mega en Due [10]

Het filterprogramma gebruikt 11 kB opslaggeheugen en 3 kB SRAM. Na een ruwe schatting voor het benodigde geheugen voor het filterprogramma inclusief het regelprogramma, komt dit ongeveer uit op op 22 kB opslaggeheugen en 6 kB SRAM (gekende waarden maal twee).

Uiteindelijk werd er gekozen om de Arduino Due te gebruiken wegens het grootste interne geheugen en de krachtigste chip [10], [11].

## 2.5. Filtering

Om een goede regeling te bekomen moet de sensordata gefilterd worden [12]. Er bestaan verschillende soorten filters om dit te realiseren. Na een literatuurstudie werd er gekozen voor een low-pass filter wegens de eenvoudige implementatie, lage benodigde rekenkracht en de effectiviteit voor deze toepassing. Een low-pass filter filtert een deel van de ruis op een signaal waardoor er een betere regeling bekomen wordt [13].

Concreet werd er gekozen voor een eerste orde low-pass filter. Hoewel een tweede orde low-pass filter de data beter filtert, vereist deze meer rekenkracht en induceert een grotere vertraging tussen het inkomende signaal en het gefilterde signaal<sup>2</sup> [12].

Formule 2.4 beschrijft de transferfunctie van een low-pass filter [12].

$$F = \frac{\omega_0}{s + \omega_0} \quad (2.4)$$

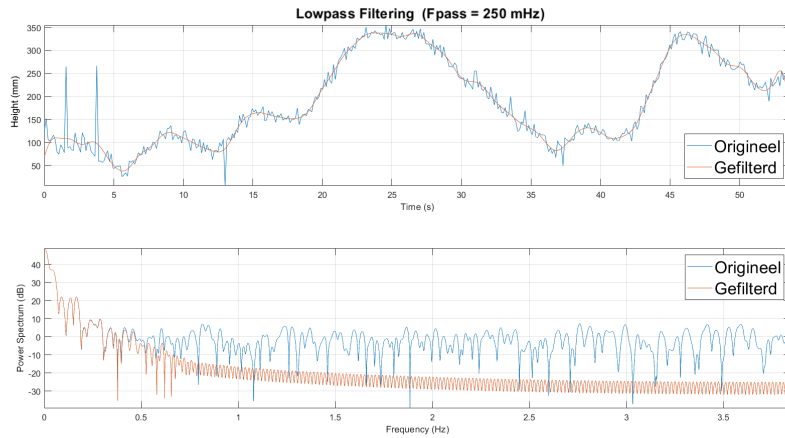
Uit formule 2.4 valt af te leiden dat een eerste orde low-pass filter één bepalende parameter heeft, namelijk de drempelfrequentie. Deze drempelfrequentie bepaalt vanaf welke frequentie de filter de signalen begint te dempen. Om deze frequentie te bepalen werd de hoogte van de boot ten opzichte van het wateroppervlak gemeten door een ultrasone sensor. Deze data werd in Matlab geïmporteerd om de juiste drempelfrequentie te bepalen. In Matlab werd de functie 'lowpass()' gebruikt. Deze functie eist, buiten de inputdata en de drempelfrequentie, een sample frequentie.

Aangezien de sample frequentie niet vast is voor een Arduino, werd een testprogramma, met een testfilter, gemaakt en werd daaruit een gemiddelde sample frequentie bepaald. Formule 2.4 geldt voor een continu systeem. In de uitgebouwde code werd ook rekening gehouden met de variërende sample frequentie<sup>3</sup>.

Onderstaande figuur (2.3) toont de ruwe data en gefilterde data van de sensoren inclusief de versterking van de verschillende frequenties:

<sup>2</sup>De grootte van de vertraging tussen een eerste orde en een tweede orde low-pass filter is afhankelijk van de drempelfrequentie, beschreven in paragraaf 2.5.

<sup>3</sup>De low-pass filter werd gebaseerd op de code van 'Curio Res' [14].



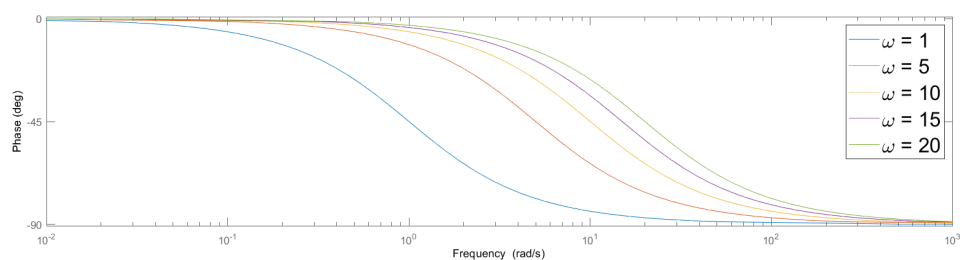
Figuur 2.3.: Eerste orde low-pass filter, drempelfrequentie 250 mHz

Boven: Hoogtemeting van de originele data (blauw) en gefilterde data (oranje) in functie van de tijd.

Onder: Versterking van de originele data (blauw) en gefilterde data (oranje) in functie van de frequentie

In de onderste grafiek (van figuur 2.3) wordt de versterking van het signaal vanaf ongeveer 250 mHz, 0 dB. De versterking wordt in de hogere frequenties ook niet significant<sup>4</sup> veel groter dan 0 dB. Dit betekent dat de signalen boven de 250 mHz ruis zijn. Deze ruis willen we onderdrukken. Om dit te realiseren kiezen we de drempelfrequentie gelijk aan 250 mHz.

Tijdens enkele testen konden we besluiten dat de filter een aanzienlijke vertraging introduceerde tussen het ingelezen en het gefilterde signaal. Deze vertraging werd kleiner wanneer de drempelfrequentie werd verhoogd. De data van deze testen werd niet opgeslagen wegens problemen het Arduino Controller Area Network (CAN) shield. De vertraging valt te verklaren uit de bode karakteristiek van de low-pass filter. Uit figuur 2.4 kunnen we afleiden dat naarmate de drempelfrequentie lager wordt, de faseverschuiving bij eenzelfde frequentie groter wordt.



Figuur 2.4.: Bode plot low-pass filter met verschillende drempelfrequenties

Door deze, eerst niet in rekening gebrachte vertraging, moet er een andere drempelfrequentie gekozen worden. Deze zal ongeveer 1 Hz zijn. Later onderzoek moet nog uitwijzen wat de exacte drempelfrequentie moet zijn waarbij er een afweging tussen de vertraging en de filtersterkte moet gemaakt worden.

<sup>4</sup>De maximum versterking vanaf 250mHz is 9 dB. De versterking van de signalen onder de 250 mHz zijn maximaal 50 dB. De nuttige data heeft een grote versterking in vergelijking met de versterking van de ruis.

## 2.6. Testen

Normaal gezien zou het volledige filterprogramma getest worden op een testvaart om zo eventueel de drempelfrequentie te verfijnen en om de geïntroduceerde vertraging van input naar de output te kunnen meten. Helaas waren er enkele complicaties op de testvaart waardoor deze metingen zijn mislukt.

Ten eerste werd er gebruikt gemaakt van een 'Arduino CAN-BUS Shield v2' om data logging mogelijk te maken. Deze was nog beschikbaar in de voorraad van het team. Dit shield is niet compatibel met de Arduino Due of met de Arduino Mega, waardoor een minder krachtige Arduino Uno moest worden gebruikt. Deze microcontroller beschikt over te weinig opslaggeheugen om het volledige filterprogramma te kunnen uitvoeren (zoals beschreven in paragraaf 2.4). Daarom werd er gekozen om het programma op slechts één hoogtesensor uit te testen in plaats van op de twee hoogtesensoren en de gyroscoop. Het volledige filterprogramma is wel uitgeschreven en klaar om in de volgende jaren gebruikt te worden<sup>5</sup>.

Een tweede complicatie ontstond tijdens de testvaart zelf. De voedingsspanning vanuit het batterijpakket was onderbroken waardoor er geen enkele test kon worden uitgevoerd.

Voor testen in de volgende jaren, wordt er aangeraden om krachtigere microcontrollers te gebruiken die compatibel zijn met CAN shields.

## 2.7. Bedrading

De sensoren werken op een spanning van 24V. De Arduino werkt op een spanning van 12V. Daarom wordt er gebruik gemaakt van twee DC naar DC omvormers. De schakeling wordt beveiligd door een zekering. De hele schakeling kan aan en uit worden gezet door een schakelaar. Zie appendix A.3 voor het volledige elektrische schema.

---

<sup>5</sup>Dit programma is te vinden in het portfolio van deze bachelorproef.

## 3. Regeling

### 3.1. Inleiding

Om een boot stabiel te laten foilen moeten er, zoals reeds vermeld in paragraaf 2.1, twee parameters geregeld worden; de hoogte van de boot ten opzichte van het wateroppervlak en roll van de boot ten opzichte van een vast referentiekader.

De hoogte moet constant blijven zodat de vaarkwaliteit zo hoog mogelijk is. Deze hoogte moet bepaald worden aan de hand van de hoogte van de golven zodat de romp zeker geen water raakt. De roll hoek van de boot ten opzichte van het wateroppervlak moet nul zijn zodat de boot horizontaal blijft. Deze bachelorproef beschouwt alleen een recht varende boot die op een constante snelheid van 30 km/u vaart<sup>1</sup>. Wanneer de boot bochten maakt, moet de roll van de boot variëren. Paragraaf 4.2 beschrijft een mogelijke methode om dit te implementeren.

Om deze parameters te regelen, moeten de aanvalshoeken van de draagvleugels aangepast worden zodat zij meer of minder lift creëren. De boot heeft drie hydrofoils, twee vooraan die regelbaar zijn en één achteraan die vast staat.

Er werd gekozen om PID-regelaars te gebruiken<sup>2</sup>. Omdat deze regelaars drie parameters per regelaar hebben (en er twee gebruikt worden), kunnen de waarden van deze parameters moeilijk handmatig bepaald worden. Daarom worden er verschillende methodes onderzocht om de ideale parameters te bepalen.

Vanuit deze waarden kan dan in de toekomst een extra handmatige tuning toegepast worden voor extra eisen rond veiligheid en behendigheid van de boot.

### 3.2. Model

Om in Simulink een regelaar uit te bouwen is er een model nodig van de boot die aan de hand van inputs, de outputs kan berekenen. (Zie figuur 3.1 voor een Simulink schema van de uitgebouwde regelaar.)

Er zijn twee inputs van het model, namelijk de twee momenten die worden uitgeoefend op de keermechanismes (zie appendix figuur: A.1) die de hoek van de foil bepalen.

De drie outputs worden bepaald door de sensordata en zijn dus de hoogte, de roll en de pitch. Dit systeem is een MIMO systeem (Multiple Inputs Multiple Outputs).

De snelheid, de yaw en de pitch worden niet geregeld. Deze worden manueel bepaald door respectievelijk: de gashendel, het roer en de snelheid.

Simulink is enorm compatibel met 'state-space models' [15]. Dit zijn wiskundige modellen die de relatie tussen de inputs en outputs uitdrukken aan de hand van de staat van het systeem [16]. Toch is er gekozen om dit model niet te bepalen omdat het enorm tijdrovend is in het bestek van deze bachelorproef en er niet verwacht werd accuraat genoeg te zijn wegens vereenvoudigingen die gemaakt moesten worden. Zo bestaat er onder andere, wegens gebrek van informatie van de oud-teamleden, geen goede wiskundige beschrijving van de huidige romp. Bovendien zijn de

<sup>1</sup>Hoewel de boot in de realiteit aan een variërende snelheid zal varen, werd er voor een eerste benadering voor een constante snelheid gekozen.

<sup>2</sup>De motivatie voor de keuze van deze regelaars wordt in paragraaf 3.3.2 besproken.

foils, tijdens het maken van deze bachelorproef, nog niet geïnstalleerd waardoor er geen validatie van het model kon worden uitgevoerd. In verder onderzoek kan er opnieuw overwogen worden om dit model op te stellen.

Een alternatieve manier om een model op te stellen, kan door Simulink een autotune proces te laten doorlopen op het fysieke systeem. Dit kon helaas ook niet gebruikt worden aangezien het foilingsysteem (inclusief de motoren) nog niet gebouwd werd. Deze methode om de regelaar in te stellen, moet later nog onderzocht worden.

Uiteindelijk werd er gekozen om een 'Motion' simulatie uit 'Siemens NX' te exporteren naar Simulink. In het Siemens NX model zijn de outputs en inputs juist omgekeerd aan de inputs en outputs in het Simulink model. Dit model werd gemaakt door Seppe Van den Broeck (in het academiejaar 2022-2023 als onderdeel van zijn masterproef) om de karakteristieken van het mechanisch geregelde foilingsysteem via simulatie te bepalen.

#### 3.2.1. Siemens NX Motion - Simulink

Het exporteren vanuit Siemens NX Motion naar Simulink wordt 'co-simulatie' genoemd. Om deze co-simulatie te kunnen realiseren werd het originele model aangepast zodat deze simulatie niet meer afhankelijk is van het mechanische regelsysteem. Vervolgens werden de twee inputs en de twee outputs gedefinieerd. Tenslotte werd het model geëxporteerd naar Simulink om het daar te implementeren met het uitgebouwde regelsysteem.

## 3.3. Regelaar

### 3.3.1. Open lus - Gesloten lus

De eerste keuze die men moet maken bij het ontwerpen van een regelaar is de keuze tussen een open lus en een gesloten lus regelaar. Van deze keuze hangen de actuatoren en de types regelaars af.

Een open lus systeem geeft geen feedback van de huidige positie van het systeem, maar regelt op basis van vooraf bepaalde parameters. Hierbij moet het gedrag van het systeem accuraat op voorhand bepaald worden. De regelaar voor een open lus systeem is doorgaans goedkoper.

Een gesloten lus systeem geeft wel feedback over de huidige staat van het systeem. Hierdoor kan de regelaar constant bijregelen als de staat van het systeem nog niet overeen komt met de gewenste staat. [17]

Er werd gekozen om een gesloten lus regelaar te bouwen omdat het gedrag van de foilende boot zeer moeilijk accuraat te voorspellen valt. Er zijn namelijk te veel onbekende invloeden zoals de wind, de stroming en eventuele fouten in computermodellen.

### 3.3.2. Types regelaar

Er zijn verschillende types regelaars waarvan de ene al meer geschikt is voor de flight controller dan de andere. Onderstaande tabel (3.1) geeft een overzicht weer van de verschillende types regelaars met hun voor- en nadelen. De score werd zelf bepaald door de voor- en nadelen af te wegen.

| Regelaar | Score | Voordelen  | Nadelen   |
|----------|-------|--|---|
| PID      | 1     | Eenvoudig, minste rekenkracht nodig en voorkennis in het team                        | Meerdere regelaars in stellen moet via iteratie en kan instabiel worden                             |
| LQR      | 2     | Weinig kans op instabiele situaties, middelmatige rekenkracht nodig                  | Niet implementeerbaar met de co-simulatie   |
| MPC      | 3     | Weinig kans op instabiele situaties, fysieke grenzen worden in de regelaar ingegeven | Meeste rekenkracht nodig, berekeningen kunnen divergeren, niet implementeerbaar met de co-simulatie |

Tabel 3.1.: Voor- en nadelen van verschillende types regelaars [18]

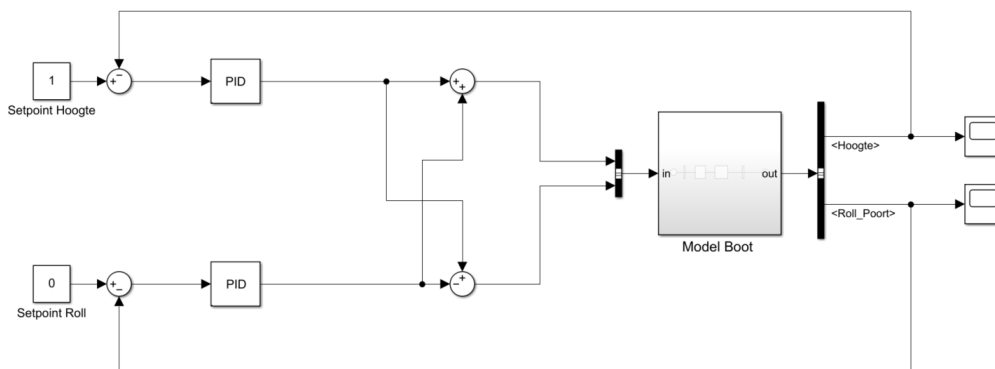
### 3.3.3. Ontwerp regelaar

Het ontwerp van de regelaar werd gebaseerd op verschillende werken uit de literatuur in samenwerking met verschillende teamleden [5], [19].

Het regelsysteem uit figuur 3.1 regelt twee parameters; de hoogte en de roll van de boot. De bovenste ingang in het model van de boot is de rechter foil en de onderste ingang is de linker foil.

Om de hoogte van de boot te regelen, moeten beide foils dezelfde hoek behouden. Daarom wordt de uitgang van de bovenste regelaar positief, met dezelfde waarde in elke input van de foils gestuurd.

Om de roll van de boot te regelen, moet de ene foil een positieve hoekverdraaiing krijgen en de andere een negatieve hoekverdraaiing. De liftkrachten van de foils zijn lineair in functie van hoek waaronder de foil zich bevindt (bij de toegepaste snelheden) [20]. Daarom wordt het hoekverschil niet geschaald volgens een functie.



Figuur 3.1.: Regelsysteem: Hoogte en Roll [5], [19]

## 3.4. Parameters bepalen

Er zijn verschillende methodes om de parameters van de regelaars te bepalen. In de onderstaande tabel (3.2) staan enkele methodes met hun voor- en nadelen weergegeven. De score werd zelf bepaald door de voor- en nadelen af te wegen.

| Method                             | Score | Voordelen  | Nadelen   |
|------------------------------------|-------|--|---|
| Simulink: Transfer Function Based  | 2     | Veel documentatie, visueel en werkt goed voor lineaire systemen      | Kan niet geïmplementeerd worden met de co-simulatie |
| Simulink: Frequency Response Based | 3     | Implementatie met co-simulatie mogelijk                              | Niet toepasbaar in een onstabiel model              |
| Heeds                              | 4     | Kan omgaan met errors en is efficiënt                                | Geen implementatie van Simulink mogelijk            |
| Iteratie                           | 1     | Kan geïmplementeerd worden en omgaan met errors                      | Omslachtig, veel rekentijd                          |
| Particle Swarm Optimization        | 1     | Kan geïmplementeerd worden, omgaan met errors en accurate resultaten | Veel rekentijd                                      |

Tabel 3.2.: Voor- en nadelen van verschillende methodes om parameters te bepalen [21], [22], [23]

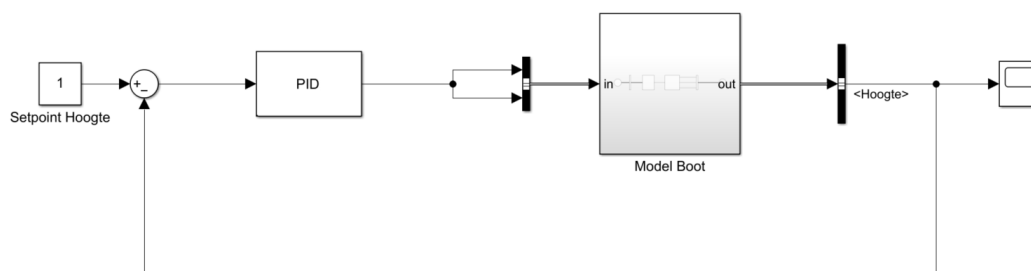
In de Simulink Control Design Toolbox zijn er nog andere methodes om een regelaar te tunen. Hier werd niet verder op ingegaan wegens te weinig kennis binnen het team en om de werklast van dit onderzoek binnen de werklast van een bachelorproef te houden.

In de toekomst moeten de andere methodes ook bestudeerd worden.

Uiteindelijk werd er gekozen om iteratie toe te passen om een idee te krijgen in welke regio de beste PID-waarden zich bevinden. Vervolgens werd de Particle Swarm Optimization (PSO) methode toegepast om de resultaten te verfijnen.

### 3.4.1. Iteratie

Bij de iteratie simulatie werden een reeks random waarden, binnen een bepaald bereik, voor elke parameter gekozen en worden de resultaten berekend. Om het algoritme eenvoudig te houden, werd in de eerste implementatie gekozen om de roll-as vast te zetten en dus niet te regelen. Hierdoor krijgen we volgend regelsysteem waar alleen de hoogte geregeld wordt:



Figuur 3.2.: Regelsysteem: Hoogte

Dit is een SISO (Single Input Single Output) systeem. Er moeten drie parameters bepaald worden, namelijk de P-, I- en D-waarde. Het setpoint staat op één, wat betekent dat de boot in de simulatie ten opzichte van een referentie één meter hoog moet foilen. Deze setpoint zal in het onderzoek van de volgende jaren moeten aangepast worden naargelang de golfhoogte, hoogte van de struts en stabiliteit.

Om de verschillende resultaten met elkaar te vergelijken werd de som van de gekwadraterde

fouten (SSE; Sum of Squared Errors) genomen. De combinatie van de P-, I- en D-waarde met de laagste SSE kiezen we als optimale parameters. De simulatietijd werd ingesteld op 150 seconden met een stapgrootte van 0.02 seconden. Er werden geen golven gesimuleerd. De gewenste hoogte werd op 0.7 m ingesteld. Onderstaande tabel 3.3 geeft vier willekeurige resultaten uit de iteratiemethode weer. De eerste rij van deze tabel (3.3) geeft het beste resultaat weer. Een SSE met waarde: NaN (Not a Number), betekent dat de simulatieresultaten divergeerden en er geen SSE kon gevonden worden.

| <b>P</b> | <b>I</b> | <b>D</b> | <b>SSE</b>               |
|----------|----------|----------|--------------------------|
| 0.06     | 0.03     | 0.02     | 419.2934                 |
| 0.01     | 0.03     | 0.02     | 420.4092                 |
| 0.33     | 0.50     | 0.50     | $154.1210 \cdot 10^{12}$ |
| 0.22     | 0.28     | 0.50     | NaN                      |

Tabel 3.3.: Resultaten PID-waarden na iteratie

### 3.4.2. Particle Swarm Optimization

De resultaten na iteratie kunnen verfijnd worden door een PSO toe te passen. In het PSO algoritme, maken we een 'zwerm' die bestaat uit verschillende 'deeltjes'. Elk deeltje in deze zwerm heeft een P-, I- en D-waarde. Hieruit wordt dan telkens de SSE berekend. Elk deeltje kent de parameters die zijn eigen minimale SSE bepalen en de parameters van de globale SSE. In elke stap krijgt elk deeltje een 'snelheid' die wordt bepaald uit zijn vorige snelheid, de parameters van zijn eigen minimale SSE en de parameters van de globale minimale SSE. Deze snelheid, samen met zijn huidige parameters, bepaalt de nieuwe parameters van het deeltje. Het is niet zeker dat deze heuristische methode een globaal minimum vindt, maar de resultaten komen vaak dicht in de buurt van dit minimum<sup>3</sup> [25]. De instellingen voor deze optimalisatie-techniek worden in volgende tabel (3.4) weergegeven:

|                                |      |
|--------------------------------|------|
| <b># Parameters</b>            | 3    |
| <b># Deeltjes</b>              | 10   |
| <b># Iteraties</b>             | 100  |
| <b>Minimale waarde</b>         | 0    |
| <b>Maximale waarde</b>         | 0.07 |
| <b>Snelheid dempingsfactor</b> | 0.99 |

Tabel 3.4.: Instellingen gebruikt bij de PSO-techniek die bepaald werden door trial and error en door [26]

De PSO-techniek werd in Matlab geïmplementeerd<sup>4</sup> en gaf de resultaten in onderstaande tabel (3.5).

<sup>3</sup>Meer informatie over de PSO-techniek: [24]

<sup>4</sup>Het PSO algoritme werd gebaseerd op de code van 'Yarpiz' [26].

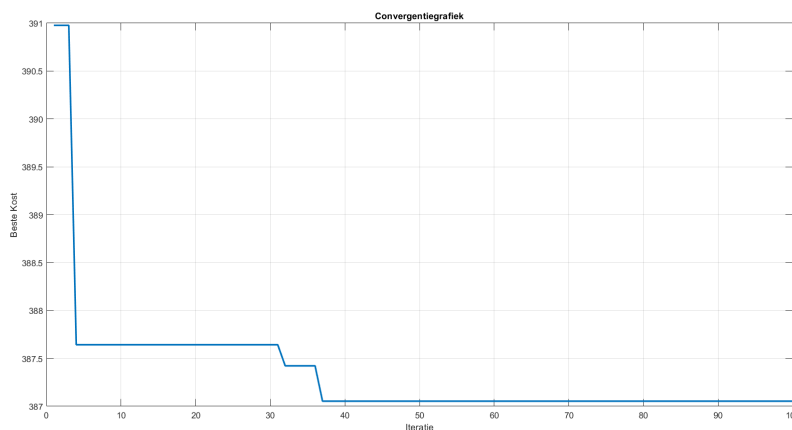


### 3.4. PARAMETERS BEPALEN

| Nummer deeltje | P      | I      | D      | SSE      |
|----------------|--------|--------|--------|----------|
| 1              | 0.0400 | 0.0361 | 0.0090 | 26307    |
| 2              | 0.0352 | 0.0314 | 0.0164 | 395.9624 |
| 3              | 0.0360 | 0.0311 | 0.0158 | 398.8269 |
| 4              | 0.0368 | 0.0312 | 0.0165 | 399.2844 |
| 5              | 0.0352 | 0.0313 | 0.0165 | NaN      |
| 6              | 0.0352 | 0.0313 | 0.0180 | 398.5089 |
| 7              | 0.0346 | 0.0313 | 0.0166 | 636.3612 |
| 8              | 0.0273 | 0.0403 | 0.0199 | 32411    |
| 9              | 0.0352 | 0.0314 | 0.0164 | 393.5080 |
| 10             | 0.0351 | 0.0314 | 0.0167 | 398.2139 |
| Globaal beste  | 0.0352 | 0.0313 | 0.0313 | 387.0532 |

Tabel 3.5.: Resultaten PID-waarden na PSO

Grafiek 3.3 toont aan dat de de resultaten uit tabel 3.5 convergeren naar één beste oplossing.



Figuur 3.3.: Convergentie grafiek PSO: beste SSE (Kost) in functie van het iteratienummer

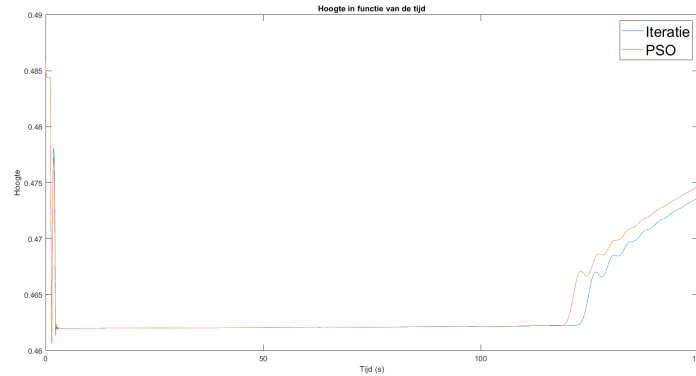
#### 3.4.3. Resultaten

Volgende tabel 3.6 geeft de beste resultaten weer voor de parameters van de regelaars na de iteratie en de PSO:

| Methode  | P      | I      | D      | SSE      |
|----------|--------|--------|--------|----------|
| Iteratie | 0.06   | 0.03   | 0.02   | 419.2934 |
| PSO      | 0.0352 | 0.0313 | 0.0313 | 387.0532 |

Tabel 3.6.: Beste resultaten PID-waarden na iteratie en PSO

We kunnen concluderen dat de PSO, zoals vooropgesteld, effectief voor een verfijning van de resultaten heeft gezorgd. Figuur 3.4 toont de hoogte van de boot in functie van de tijd bij zowel de parameters bekomen door iteratie als PSO. De regelaar met de PSO-resultaten reageert het snelste (wat beter is). Toch reageert deze regelaar enorm traag en zelfs na 150 seconden is de gewenste hoogte van 0.7 m niet bereikt.



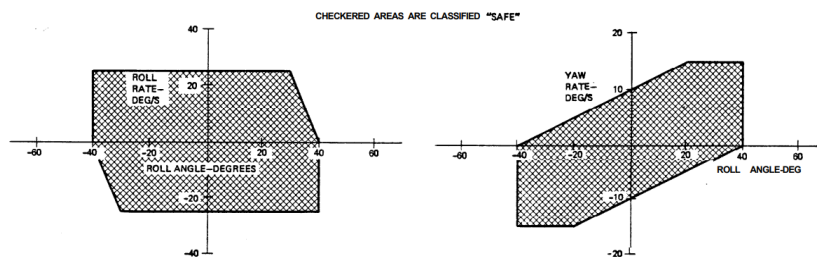
Figuur 3.4.: Hoogte in functie van de tijd met de parameters van iteratie en PSO

We kunnen besluiten dat de parameters van deze regelaars, hoewel zij de laagste SSE van elke methode hebben, niet functioneel zijn. Tijdens het simuleren werd er opgemerkt dat de co-simulatie enorm instabiel is. Eenmaal het moment op een keermechanisme te groot werd, divergeerden de resultaten. Hierdoor kan een snelle regeling niet bekomen worden.

### 3.5. Veiligheid

Tijdens het hydrofoilen kunnen er verschillende zaken mislopen; de boot kan omslaan, de foils kunnen te snel bewegen en de foils kunnen een stuursignaal krijgen dat fysisch niet mogelijk is.

De literatuur beschrijft een maximale roll en pitch hoek en versnelling tijdens het foilen om een veilige 'vlucht' te verzekeren. Deze waarden moeten nog gecontroleerd worden in toekomstig onderzoek en eventueel gecorrigeerd worden. Zij bieden wel al een goede basis om van te vertrekken.



Figuur 3.5.: Veiligheidsmarges: roll [27]

- 40 tot 40 graden roll met -25 tot 25 graden per seconde rollsnelheid
- 40 tot 40 graden roll met -15 tot 15 graden per seconde yawsnelheid

Deze veiligheidsvoorschriften moeten nog geprogrammeerd worden. Er wordt aangeraden om, wanneer de foils te snel bewegen, een veilige stop uit te voeren. Hierbij zullen de foils afremmen naar een standaardpositie waarna ze, na bevestiging van de piloot, de actieve regeling weer wordt opgestart.

Om de bewegingssnelheid en de bewegingsvrijheid van de foils te beperken, moet er een verzadigingscurve in de flight controller geprogrammeerd worden. Deze moet ook uitgetest worden in testvaarten.

## 4. Conclusie

### 4.1. Besluit

Uit dit onderzoek, naar de elektronische regeling van de hydrofoils van de UA Antwerp Solar Boat, kunnen verschillende besluiten worden getrokken.

Ten eerste kunnen we besluiten dat een low-pass filter een geschikte filter is om ruis te onderdrukken. Verder onderzoek moet nog uitwijzen wat de optimale drempelfrequentie is (rekening houdend met genoeg filteren en tegelijk de vertraging te beperken). We kunnen wel reeds stellen dat deze drempelfrequentie rond de 1 Hz zal liggen.

Ten tweede moet de keuze van de gekozen Arduino heroverwogen worden. De gepaste microcontroller moet compatibel zijn met een CAN netwerk (of een indirect via een CAN shield), moet veel rekenkracht bezitten en moet energiezuinig en modulair zijn. Een goede optie zou de Teensy 4.1 kunnen zijn [19]. Afstappen van het Arduino platform, naar bijvoorbeeld het Raspberry Pi platform, is ook een optie die overwogen moet worden.

Ten derde kunnen we besluiten dat de co-simulatie tussen Siemens NX Motion en Simulink niet ideaal is voor deze toepassing. Deze integratie bleek te instabiel en omslachtig. Er wordt voor toekomstig onderzoek aangeraden om toch een wiskundig model of een Simulink model van de hydrofoillende boot uit te bouwen. Hierdoor kunnen bovendien meer regelaars gesimuleerd worden om zo de beste te kiezen.

Vervolgens is onderzoek naar andere regelaars essentieel. Een PID-regelaar is niet robuust genoeg. Een PID-regelaar kan namelijk, door slecht gekozen parameters, instabiele situaties niet tegenwerken. Hierdoor kan de boot hevig op en neer bewegen waardoor de foils uit het water kunnen komen en de boot neervalt. Hoewel een PID-regelaar eenvoudig is, geeft deze complicaties ook bij MIMO systemen. Daarnaast kan de opstelling van de regelaar niet ideaal zijn. Zo kan het systeem beter reageren als er twee regelaars in cascade staan. Dit moet later nog onderzocht worden.

Als een ander model van de foillende boot wordt opgesteld, dan wordt er aangeraden om onderzoek te doen naar LQR- en MPC-regelaars.

### 4.2. Toekomst

Om af te sluiten kunnen fysieke aanpassingen aan de boot helpen bij het beter regelen van de foils. Zo kunnen schakelaars op het stuur zorgen tot een variërende setpoint voor de hoogte zodat de boot steeds stabiel kan varen. Pedalen voor de piloot zouden het setpoint voor de roll kunnen aanpassen zodat de boot schuin zal hangen zodat er kleinere momenten op de struts in de bochten aangrijpen. Hierdoor zal de regelaar de roll hoek, die essentieel is in bochten, niet gelijk aan nul proberen te houden.

## Bibliografie

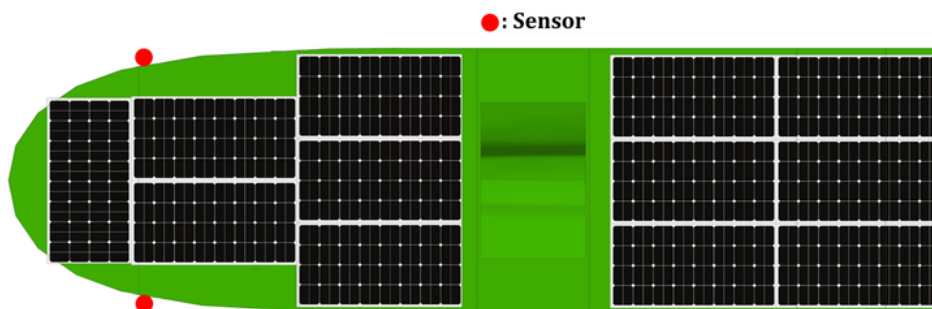
- [1] A. R. Quinteros-Condoretty, S. R. Golroudbary, L. Albareda, B. Barbiellini, and A. Soyer, “Impact of circular design of lithium-ion batteries on supply of lithium for electric cars towards a sustainable mobility and energy transition,” *Procedia CIRP*, vol. 100, pp. 73–78, 1 2021. 1
- [2] H. Babinsky, “How do wings work?” 2003. [Online]. Available: [www.iop.org/journals/physed](http://www.iop.org/journals/physed) 1
- [3] B. Liu, X. Xu, D. Pan, and X. Wang, “Research on shipping energy-saving technology: Hydrofoil amphibious vehicle driven by waterjet propulsion,” *Journal of Cleaner Production*, vol. 382, p. 135257, 1 2023. 2
- [4] M. Graf, “Mbs model for control and optimization of a hydrofoil boat,” *PAMM*, vol. 20, 1 2021. 2
- [5] J. R. Meyer and J. J. R. Wilkings, “Hydrofoil development and applications,” *Tech. Rep.*, 6 1992. 2, 3, 4, 12
- [6] “Monaco statistics,” 8 2015. [Online]. Available: <https://nl.wisuki.com/statistics/7648/monaco> 2
- [7] “Toughsonic® general purpose series ultrasonic distance sensors pc configurable and/or push-button teachable installation and operating instructions for toughsonic 3 toughsonic 12 toughsonic 14 toughsonic 30 toughsonic 50 senix corporation,” 2020. [Online]. Available: <https://senix.com/senixview-ultrasonic-sensor-software/> 5
- [8] I. Beavers, “Rarely asked questions - 139, the case of the misguided gyro,” *AnalogDialogue*, *Tech. Rep.*, 2017. 6
- [9] “Bno055 intelligent 9-axis absolute orientation sensor,” Bosh Sensortec, *Tech. Rep.*, 2014. 6
- [10] N. Anna, “Arduino uno vs mega vs due: The differences | all3dp,” 2020. [Online]. Available: <https://all3dp.com/2/arduino-mega-vs-uno-vs-due-differences/> 7
- [11] Z. Peterson, “8-bit vs. 32-bit mcu: Choosing the right microcontroller for your pcb design | altium.com,” 2 2017. [Online]. Available: <https://resources.altium.com/p/8-bit-vs-32-bit-mcu-choosing-right-microcontroller-your-pcb-design> 7
- [12] T. Hägglund, “Signal filtering in pid control,” *Ph.D. dissertation*, 3 2012. 7
- [13] K. J. Åström and T. Hägglund, *Advanced PID control*. ISA-The Instrumentation, Systems, and Automation Society, 2006. 7
- [14] “Low-pass filter,” 7 2022. [Online]. Available: <https://github.com/curiores/ArduinoTutorials/tree/main/BasicFilters/ArduinoImplementations/LowPass/LowPass2>. 0 7
- [15] “State-space models - matlab & simulink.” [Online]. Available: <https://nl.mathworks.com/discovery/state-space.html> 10

- [16] Z. Chen and E. N. Brown, “State space model,” p. 30868, 2013. 10
- [17] “What is open-loop vs closed-loop - xeryon.” [Online]. Available: <https://xeryon.com/technology/what-is-open-loop-vs-closed-loop/> 11
- [18] M. Okasha, J. Králev, and M. Islam, “Design and experimental comparison of pid, lqr and mpc stabilizing controllers for parrot mambo mini-drone,” *Aerospace*, vol. 9, p. 298, 6 2022. [Online]. Available: <https://www.mdpi.com/2226-4310/9/6/298/htmlhttps://www.mdpi.com/2226-4310/9/6/298> 12
- [19] R. Tomaz, T. Carreira, and S. Amaro, “Hydrofoil design and optimization for a solar powered boat,” Ph.D. dissertation, 2021. 12, 17
- [20] T. Putranto and A. Sulisetyono, “Lift-drag coefficient and form factor analyses of hydrofoil due to the shape and angle of attack,” *International Journal of Applied Engineering Research*, vol. 12, pp. 11 152–11 156, 2017. [Online]. Available: <http://www.ripublication.com> 12
- [21] “Open pid tuner - matlab & simulink - mathworks benelux.” [Online]. Available: <https://nl.mathworks.com/help/slcontrol/ug/designing-controllers-with-the-pid-tuner.html> 13
- [22] “Choose a control design approach - matlab & simulink - mathworks benelux.” [Online]. Available: <https://nl.mathworks.com/help/slcontrol/ug/choosing-a-compensator-design-approach.html> 13
- [23] “Frequency-response based tuning.” [Online]. Available: <https://nl.mathworks.com/help/slcontrol/ug/frequency-response-based-tuning-basics.html> 13
- [24] J. Kennedy and R. Eberhart, “Particle swarm optimization,” *Proceedings of ICNN’95 - International Conference on Neural Networks*, vol. 4, pp. 1942–1948, 11 1995. [Online]. Available: <http://ieeexplore.ieee.org/document/488968/> 14
- [25] A. Tam, “A gentle introduction to particle swarm optimization,” 10 2021. [Online]. Available: <https://machinelearningmastery.com/a-gentle-introduction-to-particle-swarm-optimization/> 14
- [26] M. K. Heris, “Video tutorial of pso implementation in matlab,” 2016. [Online]. Available: <https://yarpiz.com/440/ytea101-particle-swarm-optimization-pso-in-matlab-video-tutorial> 14
- [27] W. E. Farris, D. R. Stark, and C. T. Ray, “Hydrofoil ship control and dynamics specifications and criteria,” The Boeing Company, Tech. Rep., 1977. 16

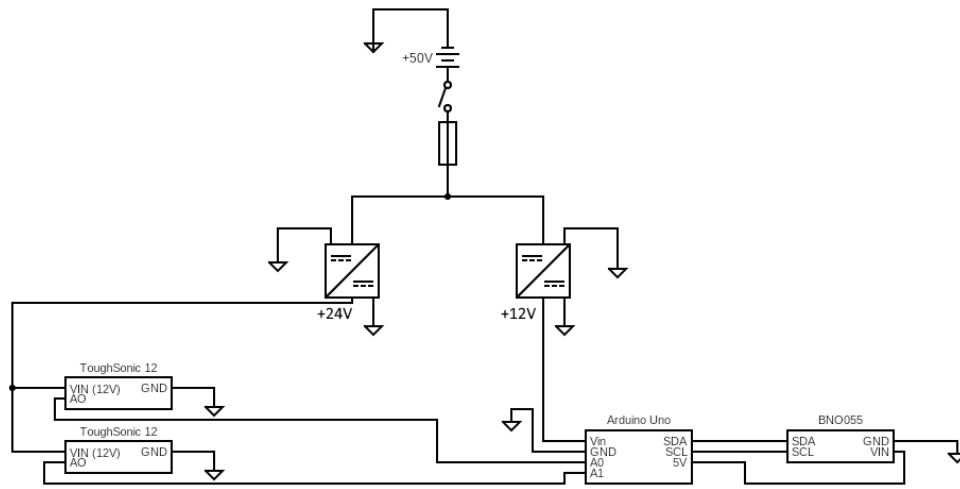
## A. Appendix



Figuur A.1.: Foil samenbouw: Keermechanisme (rood)



Figuur A.2.: Plaatsing sensoren op de boot



Figuur A.3.: Elektrisch schema flight controller, filter