

Lijnvolgsysteem voor F1/10 wagen

Anton Malonenkov

Bachelorproef tot het behalen van een diploma Bachelor of Science in de
industriële wetenschappen: elektronica-ICT

Promotor

Dr. Ing. Ken Vanherpen



Deze bachelorproef is een examendocument dat mogelijk in zijn finale versie niet werd gecorrigeerd voor eventueel vastgestelde fouten. Zonder voorafgaande schriftelijke toestemming van zowel de promotor(en) als de auteur(s) is overnemen en kopiëren van deze uitgave of gedeelten ervan verboden. Voor aanvragen tot of informatie i.v.m. het uitgeven en/of gebruik en/of realisatie van gedeelten uit deze publicatie, wendt u zich 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 gebruik en voor de inzending van deze publicatie ter deelname aan wetenschappelijke prijzen of wedstrijden. Dit document is in overeenstemming met het facultaire bachelorproefreglement beoordeeld door promotor(en) en/of begeleider(s).

Abstract

In de wereld van autonome voertuigen wordt gebruik gemaakt van verschillende methodes om het voertuig een bepaalde route op een veilige en gecontroleerde manier te laten volgen.

In dit onderzoek is een gelijkaardig systeem nageemaakt, weliswaar op een 10^{de} schaal van de werkelijkheid. Hierbij werd gebruik gemaakt enerzijds van analoge signalen gegenereerd door IR sensoren, anderzijds van *state-of-the-art* apparatuur zoals een LIDAR.

Om dit onderzoek te realiseren, is er gebruik gemaakt van verscheidene softwarepakketten en zgn. *libraries*. Onder andere Python en de Arduino code vormen de basis van het systeem, maar even belangrijk is de hardware, nl. de LIDAR, de sensoren en de Nvidia GPU (grafische kaart).

Inhoudstafel

Abstract.....	2
1 Inleiding.....	3
2 Technieken en methodes	3
3 IR.....	4
4 LIDAR	6
4.1 Input	6
4.2 Dataprocessing	8
4.3 Output (reactie van de wagen)	9
5 Conclusie	10
6 Referenties	10
7 Appendix A	11

1 Inleiding

Wanneer men het heeft over de veiligheid van auto's komt vrij snel het onderwerp van zelfrijdende wagens te pas. Denk dan in eerste instantie aan de auto's van Tesla, maar tegenwoordig ook Audi, Mercedes, Polestar, ... Deze heeft u al zeker zien rijden, of wellicht heeft u er zelf al mee gereden. Ze zijn van onder tot boven uitgerust met sensoren van verschillende types om de wagen de mogelijkheid te bieden om een uitgestippeld pad nauwkeurig te volgen.

Naast het vervoeren van mensen in dergelijke wagens, wordt de technologie ook gebruikt om zelfrijdende auto's te laten racen tegen elkaar. Om algoritmen te ontwikkelen voor zo'n toepassing, is de kwestie van geld (en dus schaal) van belang. Een volwaardige Tesla gebruiken als proefauto kan niet iedereen. Hiervoor is het platform genaamd F1/10 (F1TENTH, 2020), vandaar dus de 1/10^e schaal) ontwikkeld. Men bouwt en racet met schaalauto's die autonoom hun weg vinden op een eveneens kleiner geschaald circuit.

Het is de combinatie van deze twee onderwerpen waarover dit wetenschappelijk verslag zich buigt. Met welk type sensoren kan een systeem opgebouwd worden zodat een rc wagen een vooraf gemaakt pad volgt?

2 Technieken en methodes

Eerst en vooral wordt er uitbundig gebruik gemaakt van het F1/10 platform als bron. Daar wordt gespecificeerd wat voor chassis als basis kan gebruikt worden voor de op te bouwen wagen. Bij aanvang van dit onderzoek was het materiaal voor *Traxxas 4-Tec 2.0* wagen ter beschikking.



Figuur 1: Traxxas 4-Tec 2.0 RC Wagen

Om de wagen gecontroleerd te laten rijden op een bepaalde snelheid is een correcte instelling vereist van de VESC (*Variable Electronic Speed Controller*). Dat is een elektronisch circuit, een snelheidsregelaar, hetwelk een bepaalde input vertaalt en omzet in signalen zodat de motors op de wagen dit kunnen begrijpen. De wagen is hiervoor uitgerust met een VESC 6 Mk V van het merk Trampa. Er is beslist om de aandrijving van de motors en dus de snelheid manueel te laten gebeuren, volledig in handen van de gebruiker door middel van de afstandsbediening van de wagen. Dit als *fail-safe* mocht er iets verkeerd lopen. De VESC's van Trampa zijn in te stellen met heel diepgaande software (De vesc-tool). Met deze tool kan heel nauwkeurig de werking van de aandrijf-as worden ingesteld naarmate de wensen van de gebruiker. In dit onderzoek is het doel geweest om de wagen zo in te stellen dat de maximumsnelheid vooruit vele malen lager ligt dan een gebruikelijke rc auto. Daarnaast is de mogelijkheid om achteruit te rijden afgezet.

Op deze wagen zijn twee servomotoren aanwezig. Dat zijn kleine apparaten die een mechanisch systeem regelen op basis van elektronische signalen. Bij de wagen wordt één servo gebruikt voor de aandrijving van de VESC (vertaalt dus de input van de afstandsbediening). De andere staat in voor het controleren van de stuurwielen. Het is die tweede servo die in dit lijnvolgsysteem de laatste stap is tot een fysieke beweging van de wagen.

Men zou voor de aansturing van de eerste (snelheids-) servo ook andere elektronische apparatuur kunnen gebruiken, denk dan bijvoorbeeld aan een Teensy microcontroller in combinatie met een MicroZED.

Het voordeel van de VESC van Trampa over de Teensy is de software, met als gevolg dat de output veel stabiel is, zeker bij lage snelheden, net wat in dit onderzoek belangrijk is. Ook controleert de Trampa VESC de output veel veiliger omdat men duidelijk kan instellen wat voor batterijgrootte gebruikt wordt. Dit om de motoren niet stuk te rijden, maar ook om de batterij langer mee te laten gaan.

Om de wagen en de werking van de servo's te leren kennen werd in het eerste deel van het onderzoek de wagen uitgerust met een paar *IR sensoren*.

IR is een afkorting voor infrarood. De sensor is in staat om de weerkaatsing van infrarood licht te detecteren. Dit licht valt buiten het weerneembare spectrum van ons menselijk oog. Je gebruikt zo'n sensor wellicht bijna dagelijks, denk bijvoorbeeld aan de afstandsbediening van je tv.

Na de ontwikkeling met de IR sensoren werd overgegaan naar het uitwerken van een opstelling/geheel waarbij gebruik werd gemaakt van een LIDAR. Dat is een iets complexer stuk elektronica. Ook hier is de naam een afkorting, nl. van *Laser Imaging Detection And Ranging*.

Een LIDAR deelt zijn werkwijze met een radar: een signaal wordt uitgezonden en zal enige tijd later door reflectie weer worden geregistreerd en opgevangen. De afstand tot het oppervlak/object wordt bepaald door de tijd te meten die nodig was om het signaal terug op te vangen.

Het verschil zit bij het type signaal dat wordt uitgezonden: een LIDAR gebruikt, zoals de naam suggereert, laserlicht, terwijl een radar gebruik maakt van radiogolven.

3 IR

Het begrip IR sensor omvat twee analoge instrumenten, nl. een IR fotodiode en een IR LED.

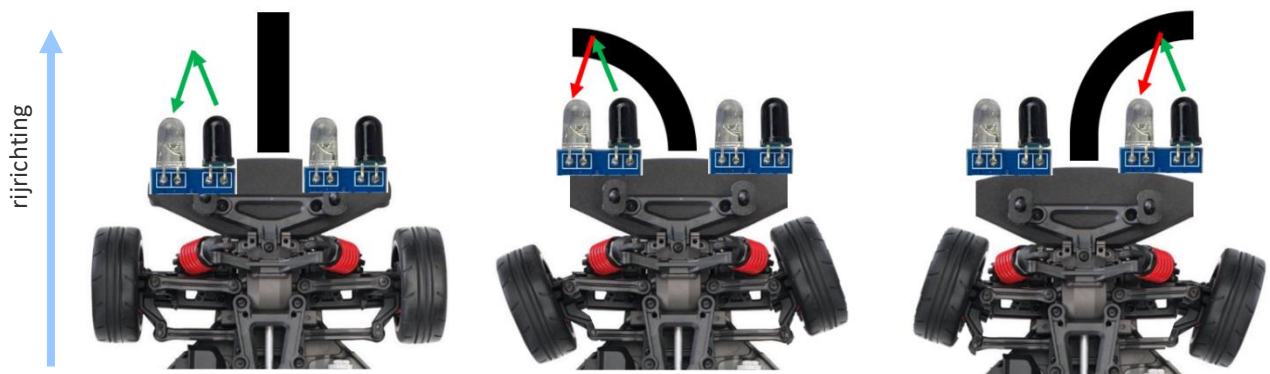
De fotodiode is gevoelig aan het licht dat wordt uitgezonden door de LED (*Light Emitting Diode*, een lichtgevende diode). Met een verschil in de waarde van het teruggekaatste licht kan waargenomen worden waar een overgang is tussen twee verschillende materialen.

De IR sensoren werden uitgelezen door gebruik te maken van een Arduino. Dat is een platform van microprocessors dat ontwikkeld is met het oog op het gemakkelijk en benaderbaar maken van apparaten en objecten, dit op hobby schaal maar ook zeker voor professionele elektrotechnici. De opstelling is vrij simpel, en ziet er schematisch uit als volgt:



Figuur 2: Schema van opstelling met IR sensoren

Om twee IR sensoren te gebruiken voor het volgen van een lijn, werd dit paar sensoren bevestigd op de wagen als volgt:



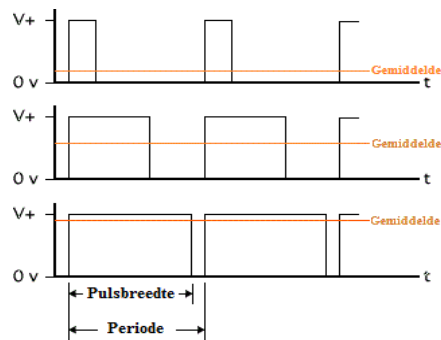
Figuur 3: Situatieschets (niet op schaal, niet in detail) toepassing van IR

Opmerking: de IR sensoren wijzen naar het oppervlak waarop de wagen rijdt, in figuur 2 is dit niet meteen duidelijk.

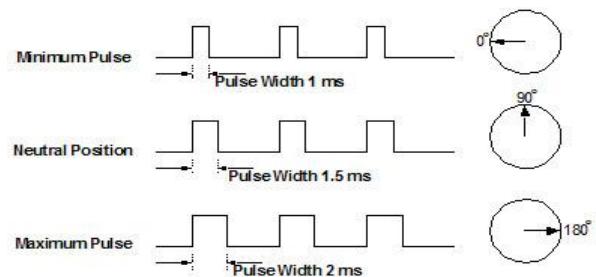
Uit het schema is wel meteen duidelijk waarom gekozen is voor twee IR sensoren. Op die manier wordt gezorgd dat de lijn die te volgen is, in het midden van de wagen blijft. In het geval dat de linkse IR sensor een aanzienlijke verlaging in gereflecteerd licht waarneemt stuurt de wagen bij naar links. Hetzelfde geldt voor de rechtse IR sensor, een detectie daar zorgt voor een bijsturing naar rechts. Zo een detectie werd geprogrammeerd op de Arduino. Daar draait een programma op dat constant de waarde van beide sensoren uitleest. Ze wordt dan vergeleken met de gekalibreerde waarde, dat is de waarde waarbij er geen lijn onder beide sensoren loopt en dus de wagen op een effen oppervlak staat. Wanneer een waarde wordt uitgelezen die verschilt van deze gekalibreerde waarde met een bepaalde drempel, wordt een signaal gestuurd op een output PWM-pin van de Arduino. Die pin is rechtstreeks verbonden met de servomotor die instaat voor de stuurwielen.

PWM is een afkorting voor *Pulse-Width Modulation* ofwel pulsbreedtemodulatie. Het is een techniek om elektrische pulsen met een vaste frequentie uit te zenden met een variërende breedte. Men stuurt een puls in de vorm van een blok golf: de spanning is dus volledig ingeschakeld ($V+$) of volledig uitgeschakeld (0 V). Bij PWM wordt de arbeidscyclus (*duty cycle*) van het signaal aangepast. Dat is de verhouding van de tijd dat het signaal hoog (1) is tot de periode van het signaal. (Pulsbreedtemodulatie, 2021) Op basis van de tijd die nodig is om een puls met die breedte te sturen, wordt bepaald met welke hoek de servomotor wordt ingesteld.

Hiervoor zijn tabellen opgesteld. (Singh, 2014) (Servo FAQ's, 2022)

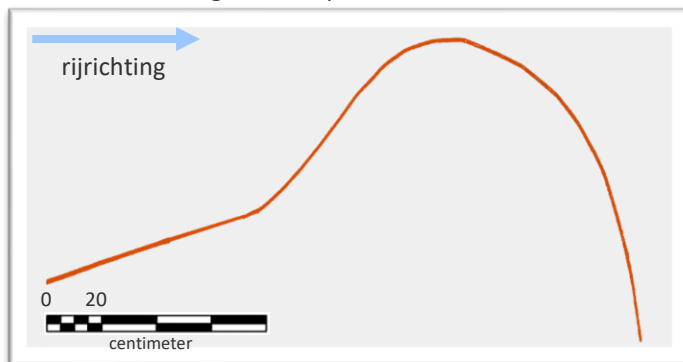


Figuur 4: Pulsbreedtemodulatie

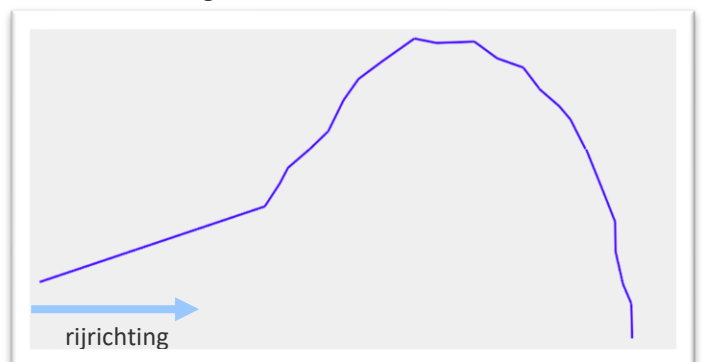


Figuur 5: Translatie van pulsbreedte naar hoek (draaiing) van de servo

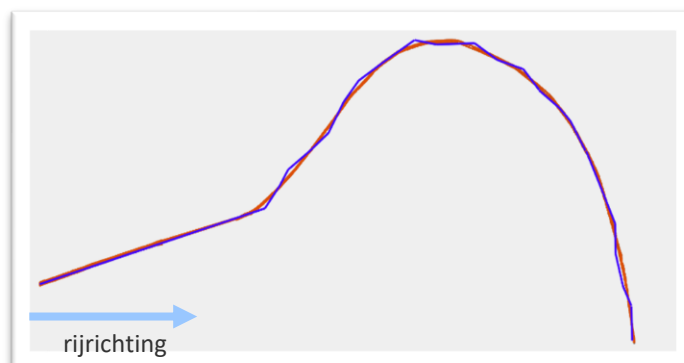
Om de werking van de gemaakte opstelling te controleren werd met zwarte elektrische tape een lijn aangebracht op de vloer. Het resultaat van de finale test is te zien figuur 7 en 8:



Figuur 6: Opgestelde lijn



Figuur 7: Beweging van de wagen op basis van IR sensoren



Figuur 8: Verschil tussen de controle lijn en het resultaat van de gevolgde lijn door de wagen

Uit de grafiek is te zien dat, met het geschreven Arduino programma, de wagen goed de controlelijn volgt.

Het grote nadeel aan het gebruiken van IR sensoren is het feit dat de wagen nagenoeg stapvoets moet rijden, om op tijd een detectie uit te voeren en goed te kunnen reageren. Ook is de keuze van materiaal voor de te volgen lijn belangrijk, het moet detecteerbaar zijn door de IR sensoren. Dit wil zeggen dat het weinig licht moet reflecteren. Daarnaast gebeurt het dat de wagen bij een foutieve detectie al snel de lijn kwijtspeelt. Het systeem, op basis van IR sensoren, is niet in staat zich te herstellen, m.a.w. de wagen kan de lijn niet zoeken en zijn traject herstellen.

Het is met die redenen dat in het tweede deel van dit onderzoek is overgeschakeld naar het gebruik van de iets complexere LIDAR.

4 LIDAR

De LIDAR die gebruikt werd voor dit onderzoek is een RPLIDAR A3, en is rechtstreeks aangestuurd door een Nvidia GPU.



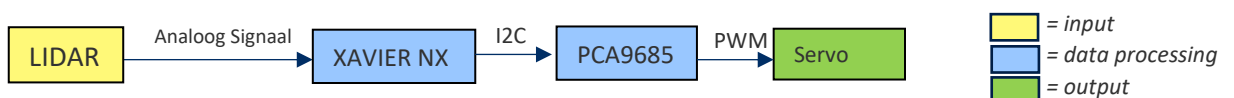
Figuur 9: RPLidar A3



Figuur 10: Nvidia Jetson Xavier NX

Daar de LIDAR tot 16000 metingen per seconde kan doen, is ook een veel snellere processor nodig dan de Arduino. Die heeft slechts een enkele processor met een maximumsnelheid van 16MHz. De Jetson daarentegen heeft een 6-core CPU met 1400-1900MHz aan maximum klokfrequentie. Daarnaast is er ook een GPU met klokfrequentie van 1100MHz. Hierdoor kunnen de metingen veel sneller verwerkt worden dan op een Arduino of dergelijke microprocessor.

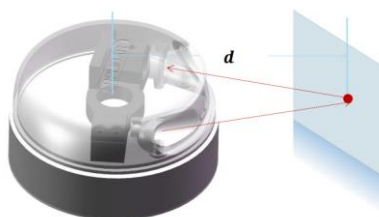
Op de Nvidia Jetson draait een versie van Linux genaamd JetPack. Om de LIDAR zijn data uit te lezen is Python gebruikt. De opstelling met de LIDAR ziet er schematisch zo uit:



Figuur 11: Schema van opstelling met LIDAR

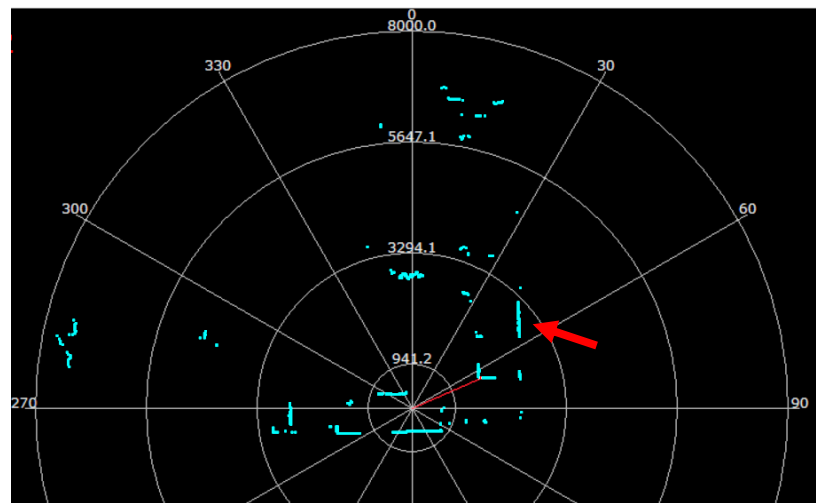
4.1 Input

De LIDAR stuurt een infrarood laser signaal uit en registreert dit een fractie van een seconde later terug op de fotosensor. De omzetting van deze data gebeurt door het RPLIDAR vision acquisition system. (ROS Components, 2016)



Figuur 12: RPLIDAR afstandsdetectie

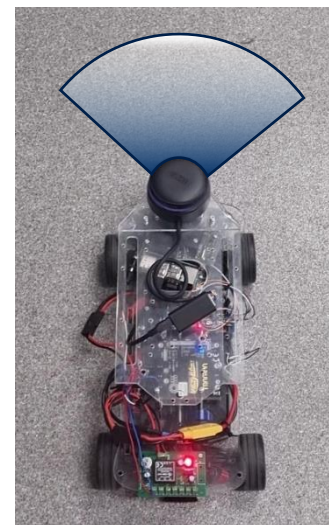
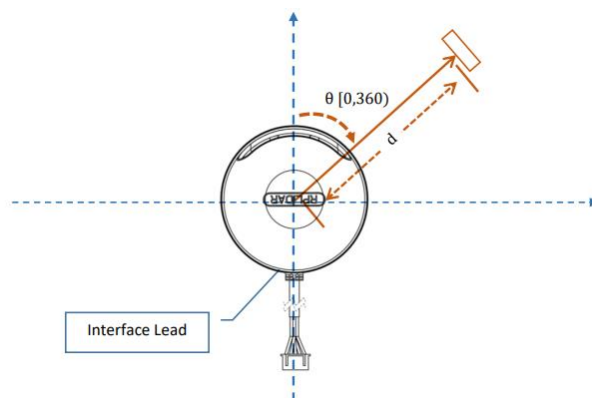
Wat men als gebruiker terugkrijgt is dan voor elke hoek de afstand tot een of ander object dat zich daar bevindt. Een voorbeeld van een detectie die gebeurde in een kleine kamer:



Figuur 13: RPLIDAR kamer detectie

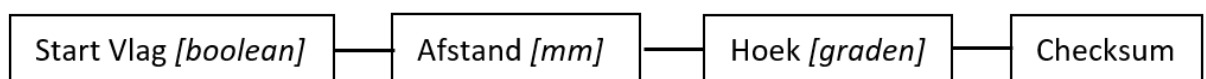
Er zijn enkele muren te zien, bv. op ongeveer 55 graden en op 2.6meter (rode pijl).

Het idee om een LIDAR te gebruiken voor een zelfrijdende auto houdt in dat de wagen moet kunnen zien of detecteren als er iets in zijn weg staat, d.w.z. dat er enkel naar voor moet gekeken worden (blauwe gradient in figuur 13). Hiervoor werd een bereik ingesteld van 0-45° en van 315-359°. Merk op dat de rotatiehoek toeneemt met de wijzers mee.



Figuur 14: Hoekrotatie en voorstelling op de wagen

Een meting van deze LIDAR ziet er structureel als volgt uit:



Figuur 15: Dataformaat van RPLIDAR-protocol

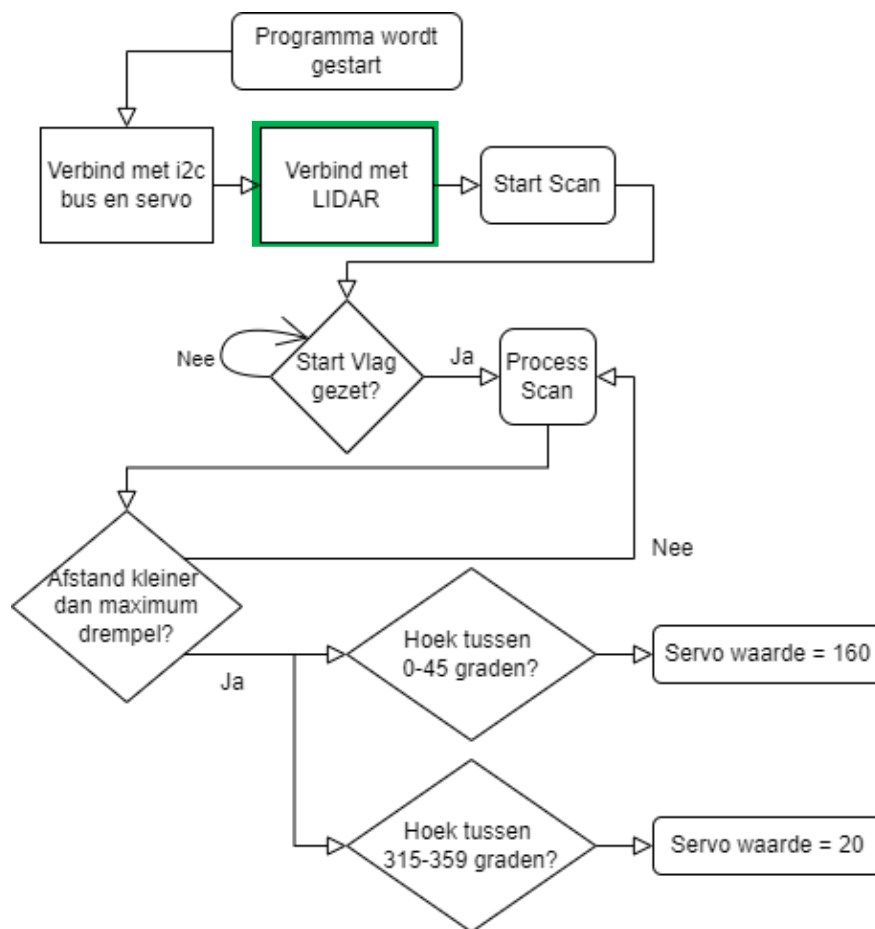
(ROS Components, 2016)

Het is vooral de afstand en de hoek waarop het python programma zich baseert.

4.2 Dataprocessing

De waarden gegenereerd door de LIDAR kunnen op verschillende manieren worden geïnterpreteerd en verwerkt. Een eerste optie in dit onderzoek was het gebruik van Robot Operating System. (ROS, 2021) Hiermee kan een mapping gemaakt worden van waar de LIDAR zich bevindt relatief tot de wagen en het oppervlak waarop hij rijdt. Met oog op de simpliciteit van het gehele systeem werd besloten om afstand te doen van ROS en de LIDAR rechtstreeks aan te spreken met behulp van Python.

Het zelfgeschreven Python programma maakt gebruik van twee heel belangrijke softwarepakketten, nl.: **pyrplidar** en **adafruit_servokit**. In dit hoofdstuk wordt meer verteld over het eerste, de servokit wordt behandeld in het volgende hoofdstuk. (Hyun, 2019) (Adafruit, 2018)
Het programma volgt de volgende logica:



Figuur 16: Flowchart Python LIDAR programma

Er wordt eerst gewacht op een succesvol gezette start vlag, op dat moment is de LIDAR zeker klaar om correcte data te sturen. Vervolgens wordt in een lus de data gelezen, als de uitgelezen afstand kleiner is dan de ingestelde maximum drempel, wordt gecontroleerd tussen welke hoeken dit object zich bevindt. Zodoende wordt bij een gedetecteerd object aan de linkse kant van de wagen (hoek tussen 315 en 359 graden) bijgestuurd naar rechts, bij een detectie aan de rechtse kant (hoek tussen 0-45 graden) wordt bijgestuurd naar links. Merk op dat de waarden voor de servo (20 en 160) niet volledig links en volledig rechts zijn. Op die manier stuurt de wagen niet al te schokkerig bij. Deze waarden zijn bekomen na verscheidene testen. Het python programma is snel genoeg om op tijd een servo signaal te sturen. Een lus waarin de scan gelezen wordt duurt gemiddeld tussen de een à zeven nanoseconden. Een omwenteling van de LIDAR van 360 graden duurt gemiddeld 0,11 seconden.

4.3 Output (reactie van de wagen)

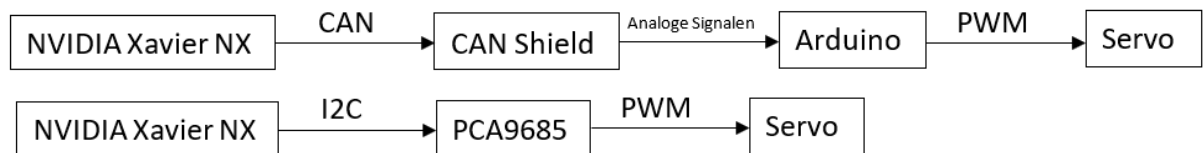
Het python script stuurt een signaal uit dat de servo moet kunnen interpreteren. Het is mogelijk om een PWM-signaal te zenden op een GPIO (General Purpose Input Output) pin van de Jetson, deze heeft er enkelen ter beschikking. Uit verscheidene testen en bronnen (Nvidia fora) werd er toch voor gekozen om dit PWM-signaal niet rechtstreeks door de Jetson te laten sturen naar de servo. Een eerste proefopstelling werd bedacht waarbij de zending van het signaal naar de servo gebeurde via een CAN BUS tussen de Jetson en de Arduino.

CAN is een afkorting voor Controller Area Network. Het is een message based protocol waarbij er in het systeem zich geen “host” bevindt. Op die manier wordt het gebruikt in elke wagen die de dag van vandaag op straat te zien is. Men verbindt elke module onderling en niet tot een “centrale” module. CAN is robust, en kan snelheden tot 1 één miljoen bits per seconde bereiken. Het python script zou dan een CAN-connectie opzetten met een CAN shield dat bevestigd werd op de Arduino. Dat CAN shield zorgt voor een differentiele connectie met de bus.

Daar er laattijdig werd opgemerkt dat de CAN pinnen op de Jetson (Transmit, Receive, V+ en grond) éérs verbonden moeten worden met een CAN Transceiver, is er in dit onderzoek niet verder gegaan met de ontwikkeling van de CAN-bus connectie.

Er is gebruik gemaakt van een *servo-control board* (PCA9685). Hiermee is het mogelijk om de Jetson een signaal te laten sturen via i2c, naar het control board. Tegenover CAN heeft I2C zijn nadelen/voordelen, zoals elke vergelijking tussen datatransmissie protocollen. I2C is doorgaans trager dan CAN, maar dit is verwaarloosbaar omdat de afstand tussen de Jetson en het controle board minder is dan 15-20 centimeter. Een voordeel van I2C is wel dat het standaard is ingebouwd in elke Jetson Xavier en dus zonder setup te gebruiken is. Dit servo control board wordt ook aangeraden om te gebruiken. Stel dat in een latere fase de aansturing van de snelheid van de wagen toch automatisch wordt gedaan, kan dit eenvoudig door ook die servo aan te sluiten op dit control bordje.

Daarnaast is er door het gebruik van dit bordje minder overhead en dus minder plekken waar het systeem kan falen. Een vergelijking tussen de twee opstellingen:



*Figuur 17: (Boven) Originele opstelling transmissie van servo signaal van Jetson naar Servo
(Onder) Nieuwe opstelling transmissie van servo signaal van Jetson naar Servo*

In figuur 16 is uitgelegd dat er eerst een verbinding wordt gemaakt tussen de Jetson en het I2C servo control board. Dat is mogelijk door gebruik te maken van de SDA en SCL pinnen die ter beschikking zijn op de Jetson. Wanneer een succesvolle connectie is gemaakt tussen de slave (het control board) en de master (de Jetson) kunnen vanuit Python, d.m.v. het pakket **adafruit_servokit**, rechtstreeks servomotoren worden aangestuurd. Het is dus dit controle board dat de vertaling maakt van i2c data naar een PWM-signaal dat de servo kan verstaan.

Om deze opstelling te toetsen op nauwkeurigheid werd net zoals bij de opstelling met IR sensoren, een testplaats opgesteld. In appendix A is het resultaat te zien van deze test, waarbij de wagen zeer goed te muren heeft ontweken en/of gevolgd.

5 Conclusie

In dit onderzoek zijn twee type sensoren onderzocht om een rc wagen een pad te laten volgen. Het is meteen duidelijk dat het gebruik van een LIDAR meer voordelen met zich meebrengt dan IR sensoren. Met de LIDAR is er geen lijn die de wagen kan kwijtraken, de wagen kan sneller rijden en het gebruik is universeel. Met dat laatste wordt bedoeld dat de LIDAR in eender welke omgeving zijn functie kan en zal uitvoeren, met weinig setup. Met IR sensoren is er eerst een lijn nodig die de wagen moet volgen, met LIDAR kan de wagen geplaatst worden tussen twee kokers van eender welk (niet transparant) materiaal en hij zal hiertussen navigeren. Dit is ook de reden dat bij f1/10 een LIDAR gebruikt wordt. Daar racen de wagens aan vrij hoge snelheid op een vooropgesteld circuit.

De twee besproken type sensoren zijn echter niet de enige die men hiervoor kan gebruiken. Het is mogelijk om het gebruik van AI (Artificiële Intelligentie) toe te passen samen met een 3d sensing camera, zoals een Stereolabs ZED 2 (STEREOLABS, 2022). Hiermee kan naargelang het doel een algoritme worden ontwikkeld zodat de wagen specifieke objecten zal ontwijken. (Chen, et al., 2017)

Wat eventueel zou kunnen verbeterd worden is het toevoegen van een PID-algoritme. Hiermee kan de wagen nog soepeler draaien. Dit is in dit onderzoek genoeg niet gelukt door tijdsgebrek. De manier hoe de wagen in dit onderzoek bijstuurt is door trial & error bijgeschaafd naar correcte waarden voor de stuurservo.

Het is ook zo dat met eender welk project dat grotendeels afhangt van (python) code, je heel ver kan gaan in de functionaliteit die je wilt en/of kan toevoegen. Moest men bijvoorbeeld in een verder stadium de LIDAR willen gebruiken om de wagen een kamer of magazijn te laten mappen en zich autonoom te laten bewegen (zonder invoer van een gebruiker aan de hand van een afstandsbediening), kan ROS (Robot Operating System) (ROS, 2021) hier verder voor gebruikt worden.

6 Referenties

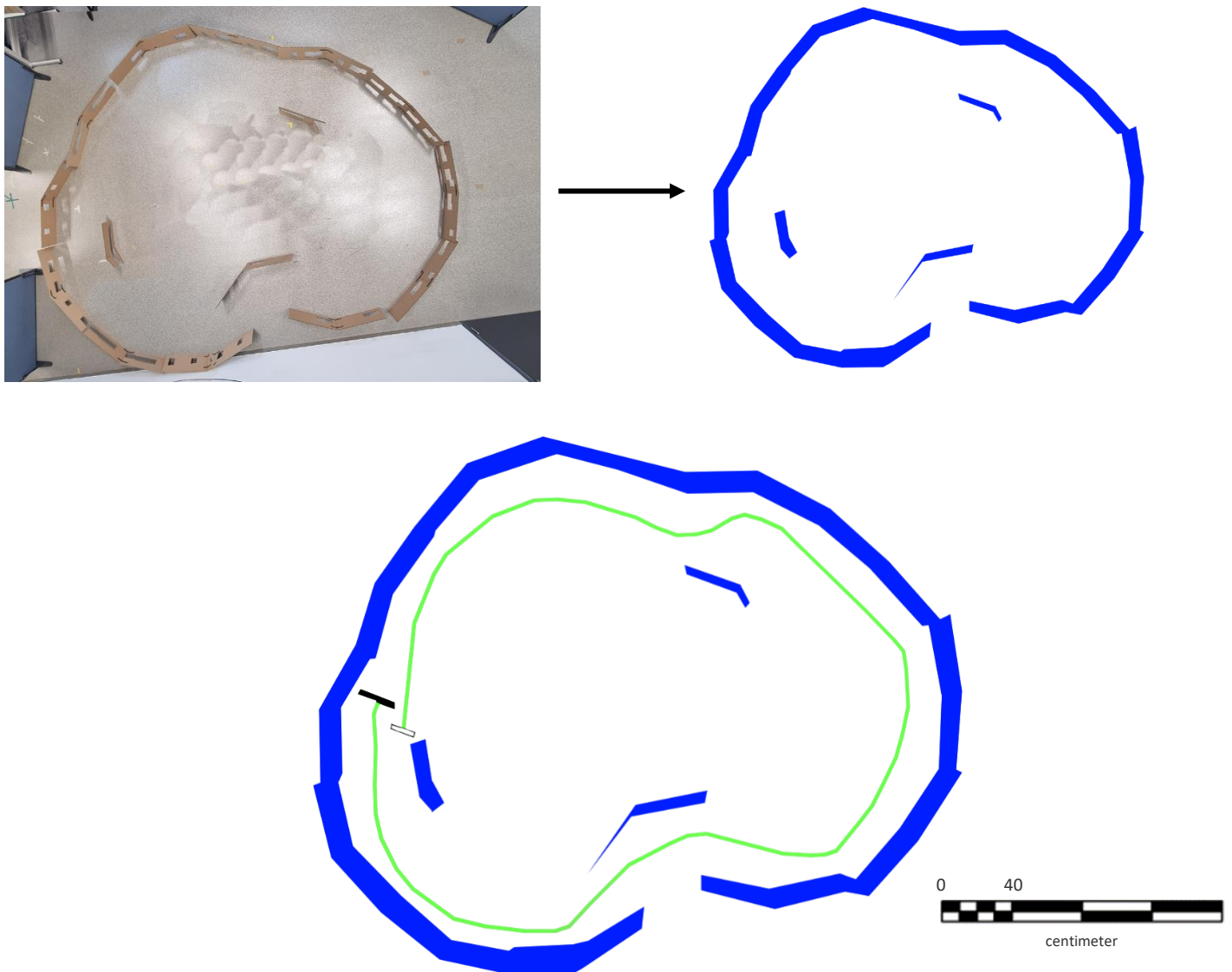
- Adafruit. (2018). *CircuitPython servokit*. Opgehaald van github:
https://github.com/adafruit/Adafruit_CircuitPython_ServoKit
- Chen, X., Kaustav, K., Yukun, Z., Huimin, M., Sanja, F., & Raquel, U. (2017, april 25). *3D Object Proposals using Stereo Imagery for Accurate Object Class Detection*. Opgehaald van arxiv:
<https://arxiv.org/pdf/1608.07711.pdf>
- F1TENTH. (2020). Opgehaald van f1tenth: f1tenth.org
- Hyun, J. (2019, june). *PyRPLidar*. Opgehaald van github: <https://github.com/Hyun-je/pyrplidar>
- Pulsbreedtemodulatie. (2021, jan 23). Opgehaald van Wikipedia:
<https://nl.wikipedia.org/wiki/Pulsbreedtemodulatie>
- ROS. (2021). Opgehaald van ROS: <https://www.ros.org/>
- ROS Components. (2016). *RPLIDAR A3*. Opgehaald van ROS Components:
<https://www.roscomponents.com/en/lidar-laser-scanner/247-rplidar-a2m8.html#:~:text=In%20enhanced%20mode%2C%20RPLIDAR%20A3,reliable%20resistance%20to%20daylight%20interference.>

Servo FAQ's. (2022). Opgehaald van servocity.com: <https://www.servocity.com/servo-faqs/#:~:text=position%20where%20the%20servo>

Singh, S. (2014). *Pulse-width modulated signal generation for control of servo motor on simplest microprocessor*. Opgehaald van Academia: https://www.academia.edu/12209755/Pulse_width_modulated_signal_generation_for_control_of_servo_motor_on_simplest_microprocessor

STEREOLABS. (2022). *Zed-2*. Opgehaald van Stereolabs: <https://www.stereolabs.com/zed-2/>

7 Appendix A



Appendix A: De wagen begon bij de in het zwart omkaderde witte rechthoek, ontweek de muren volgens het groene pad en eindigde in de volle zwarte rechthoek.