

Genereren van een kaart op basis van de verzamelde coördinaten van een autonome robot middels een webinterface

Semih Can Karakoç
Computer Science and Engineering
Inholland University of Applied Sciences
Alkmaar, The Netherlands
695258@student.inholland.nl

Samenvatting—In dit artikel is onderzocht hoe een *reverse engineered robot-vacuum* de contouren van een ruimte in kaart kan brengen door langs de muren te rijden en de coördinaten ervan naar een *webserver* te communiceren. Dit onderzoek richt zich op hoe deze webserver de gegevens kan ontvangen en de coördinaten kan verwerken met een *JavaScript Object Notation*-formaat. Verder is er onderzocht naar programmeermogelijkheden om op een object-georiënteerde wijze een *webapplicatie en -server* op te zetten waarmee de contouren van een verkende ruimte gevisualiseerd kan worden. Tenslotte is ook ingegaan op de minimaal werkend systeemontwerp (*Minimum Viable Product*), dus hoe de hard- en software resultaten verkregen zijn tijdens het onderzoek en welke verbeterpunten in een volgende release van het project aanbevolen wordt.

Index Terms—robotics, hardware, software, C#, dotnet, Raspberry Pi, access point, webinterface, webserver, webapplication, mapping, WebAssembly, SkiaSharp, API, JSON, GPIO, DBH-12V, autonome

I. INLEIDING

Het nauwkeurig in kaart brengen van de contouren van een ruimte is van cruciaal belang in diverse toepassingen van robotica, zoals autonoom navigeren, objectherkenning en uit kunnen voeren van bepaalde opdrachten (functies). In de praktijk wordt bijvoorbeeld bij de consumenten robotstofzuigers (*Samsung* was de eerste) mapping-algoritmen toegepast om de robot zijn taken meer effectiever te maken [1]. Om een ruimte in kaart te brengen spelen sensoren een grote rol. Denk aan veel toegepaste sensoren die werken met infrarode licht (IR-sensor), bumpersensoren of ultrasone geluid (ultrasone sensor) [2].

Dit artikel behandelt het ontwikkelen van een systeem, waarbij een reverse-geëngineerde besturing van een robot stofzuiger, de robot langs de muren laat navigeren van een ruimte en de coördinaten ervan communiceert naar een webserver. Deze coördinaten worden vervolgens verwerkt in een database via een webserver en door middel van een webapplicatie kunnen deze coördinaten weer verwerkt worden tot een visualisatie van de contouren van een ruimte. Bij de hardware besturing van de robot is er gebruik gemaakt van: IR-sensoren, bumper-switches, *Raspberry Pi 3B+*, motorcontroller en een eigen

ontworpen sensorhub. Er wordt eerst een minimaal werkend systeem (*Minimum Viable Product*), bestaande uit hard- en software, opgeleverd en tevens vooruitgekeken hoe de robot zijn taken nauwkeuriger en beter kan uitvoeren door toevoeging van andere sensor(en).

Een van de technische problemen die hierbij opgelost dient te worden is het continu opsturen en verwerken van de coördinaten van de route van de robot en middels een webapplicatie een kaart presenteren van de onbekende ruimte.

De relevantie van het oplossen van dit technische probleem is dat de gebruikers in staat stelt om op afstand via een *access-point* de kaart van een ruimte via een webapplicatie kan visualiseren. De toepassing in de praktijk is terug te vinden in bijvoorbeeld: huishoudrobot of grasmaairobot. Gevraagd is een OO-methode (object-georiënteerd) toe te passen bij de ontwikkeling van dit systeem.

Het doel van dit onderzoek betreft oplossen van het technische probleem met een minimaal werkend systeem (MVP). Voor dit onderzoeksartikel zijn de volgende onderdelen van methode toegepast: formele analyse, ontwerp en experiment.

A. Overzicht

De lezer zal met een gestructureerde manier door dit onderzoeksartikel begeleid worden. Zie hieronder de toegepaste structuur:

- *paragraaf II Probleemstelling*: worden de hoofdvraag en deelvragen geformuleerd;
- *paragraaf III Methoden*: worden de toegepaste methoden uitgebreid uitgelegd;
- *paragraaf IV Resultaten*: worden de resultaten van de gebruikte methode uitgewerkt;
- *paragraaf V Conclusies*: worden conclusies getrokken uit de resultaten;
- *paragraaf VI Discussie*: wordt een reflectie gedaan op de opgezette onderzoek.

- *paragraaf VII Aanbevelingen*: worden aanbevelingen gedaan voor een mogelijke vervolgonderzoek.

II. PROBLEEMSTELLING

Visualiseer de contouren van een ruimte dat door een autonome robot wordt verkent en de coördinaten ervan verwerkt worden in een database op basis van een webapplicatie en -server dat object-georiënteerd geprogrammeerd is.

A. Probleemanalyse

Het probleem kan worden beschreven als het ontbreken van een soft- en hardware ontwerp om de navigatie gegevens vast te stellen, te communiceren en te verwerken tot een kaart van de contouren van een ruimte. Technisch gezien ontbreekt de dataverwerking en de communicatie mogelijkheden om de coördinaten op te kunnen slaan in een database om vervolgens een kaart te presenteren via de webapplicatie.

Om dit mogelijk te maken zijn de coördinaten van de route van de robot nodig. Deze coördinaten zullen gecommuniceerd moeten worden via de webserver, waarna dit gebruikt kan worden voor het maken van een kaart. Daarbij speelt nog een ander probleem, dat betreft de benodigde *access-point* waarop via een *webinterface* middels een *end-device* de kaart weergegeven kan worden. Echter, om dit probleem op een simpele en doeltreffende manier te kunnen tackelen is het noodzakelijk om de verschillende programmeermogelijkheden te onderzoeken.

B. Vooronderzoek

Het doel van dit vooronderzoek is het verkennen van verschillende programmeermogelijkheden om de data van de robot te verzamelen en te verwerken om de gestelde doel te kunnen bereiken. Een ander doel is het onderzoeken naar hardwaremogelijkheden, denk bijvoorbeeld aan wat voor soorten sensoren er beschikbaar zijn en welke toegepast kunnen worden in dit project. Daarvoor zijn meerdere literatuur-vooronderzoeken gedaan, namelijk:

- Uit het eerste literatuuronderzoek [3] is gebleken dat er twee soorten traditionele database structuren bestaan, namelijk: *MySQL (Structure Query Language)* en *NoSQL*. MySQL is een relationele database en deze biedt sterke data-consistentie door middel van de ACID (*Atomicity, Consistency, Isolation, and Durability*) eigenschappen. ACID is belangrijk bij toepassingen waarbij data secuur en integer opgeslagen dient te worden.

NoSQL is een niet-relationele database en deze biedt diverse soorten manieren om data met bepaalde structuren op te slaan, in andere woorden: flexibeler qua structuur. NoSQL gebruikt BASE (*Basically Available, Soft-state, and Eventually-consistent*) eigenschappen. BASE eigenschappen worden toegepast in grote dynamische en grote databases, waarbij snelheid een grote rol speelt en consistent minder belangrijk is.

Op basis van de *benchmarks* en de technische overwegingen is NoSQL vaak sneller (op zijn snelst met een $\mathcal{O}(1)$) in het uitvoeren van basisgegevensbewerkingen ten op zichte van MySQL. Echter blijkt MySQL veel consistentier met complexere queries te zijn en is MySQL veel secuurder te zijn ten op zichte van NoSQL.

- In de tweede literatuuronderzoek [4] is een vergelijking gedaan tussen MySQL en NoSQL en welke beter is in IoT (*Internet of Things*) toepassingen. Relationale databases, zoals MySQL bieden consistentie, integriteit en veiligheid door het volgen van de ACID-eigenschappen, volgens het literatuuronderzoek. NoSQL databases, zoals MongoDB bieden daarentegen meer flexibiliteit, schaalbaarheid en (snelheid)-prestaties voor ongestructureerde en semi-gestructureerde data NoSQL volgt BASE-eigenschappen die zorgen voor uiteindelijke consistentie en hoge beschikbaarheid.

In een IoT-omgeving, waarbij *real-time* data door verschillende sensoren opgeslagen wordt, kan de keuze van de database een grote impact hebben op de prestaties van het systeem. Het literatuuronderzoek vergelijkt de prestaties van MySQL en MongoDB voor een kleine IoT toepassing: een water sprinkler systeem waarbij gebruik wordt gemaakt van diverse scenario's voor zowel datainvoer als dataopvraging. In het literatuuronderzoek worden resultaten getoond, waarin je kunt zien dat MongoDB sneller is bij het invoegen van gegevens en vooral bij een hoge *thread-count*. MySQL presteert echter beter bij leesbewerkingen onder zware belasting en blijft stabiel bij het verwerken van grote hoeveelheden gegevens. Dit suggereert dat voor IoT-toepassingen met intensieve schrijfoperaties en onvoorspelbare datastromen MongoDB toch geschikter kan zijn. Voor toepassingen die een hoge mate van data-integriteit en leesintensieve operaties vereisen, lijkt MySQL voordelig.

- De derde literatuuronderzoek [5] gaat een stap verder in op de toepassing van zowel relationele als non relationele databases. Dit literatuuronderzoek beschrijft een webapplicatie die is ontwikkeld om de evolutie van de COVID-19-pandemie in Roemenië te volgen en te voorspellen. De webapplicatie maakt gebruik van zowel relationele als non-relationele databases dat zorgt voor een optimale gegevensbeheer-strategie. Relationale databases worden gebruikt voor gestructureerde en minder volumineuze gegevens, terwijl non-relationele databases worden ingezet voor grote datasets en complexe gegevensmanipulaties.

De webapplicatie is ontwikkeld met Blazor binnen C# met een object georiënteerde wijze dat aansluit bij het onderzoeksrichting van dit artikel. Blazor stelt ontwikkelaars in staat om interactieve webapplicaties te bouwen en biedt een integratie van frontend en backend. Het gebruiken van een database in combinatie met Blazor zorgt

ervoor dat de webapplicatie zowel flexibel als schaalbaar is. Hoe deze onderzoekers Blazor hebben toegepast is een handige referentie voor dit huidige onderzoek.

- In de vierde literatuuronderzoek [6] is er onderzocht wat het verschil is tussen een LiDAR, infrarode en ultrasone sensoren voor een bepaalde hoogtemeting. Voor het meten van hoogte bij onbemande luchtvaartuigen worden verschillende sensoren gebruikt, waaronder LiDAR, infrarood en ultrasone sensoren. Deze diverse sensoren hebben unieke kenmerken die hun geschiktheid voor specifieke toepassingen kunnen bepalen.

LiDAR-sensoren (*Light Detection And Ranging*) gebruiken laserpulsen om de afstand tot een object te meten door de tijd te meten die nodig is voor het licht om terug te kaatsen. Ze bieden hoge nauwkeurigheid zelfs bij lange afstanden tot wel 40 meter met een resolutie van 1 centimeter. LiDAR is geschikt voor nauwkeurige hoogtemetingen in verschillende omgevingen en is minder gevoelig voor kleur- en textuurvariaties van objecten.

Infrarode sensoren meten de afstand door de tijd te meten die nodig is voor een infrarood lichtpuls om terug te kaatsen van een object. Deze sensoren zijn eenvoudig in gebruik en minder duur dan bijvoorbeeld een LiDAR, maar ze hebben beperkingen zoals gevoeligheid voor omgevingslicht en weersomstandigheden, zoals direct zonlicht. De maximale bereikbaarheid van infraroodsensoren varieert ongeveer van 1 tot 5 meter, maar dat is afhankelijk van het model.

Ultrasone sensoren gebruiken geluidsgolven met hoge frequentie om de afstand tot een object te meten op basis van de tijd die nodig is voor het geluid om terug te kaatsen. Deze sensoren zijn makkelijk te implementeren en hebben een bereik van enkele centimeters tot ongeveer 4 meter. Ultrasone sensoren zijn minder geschikt voor complexe omgevingen, zoals bossen vanwege hun gevoeligheid voor geluidsreflecties van ruimtelijke natuur.

- In de vijfde literatuuronderzoek [7] is er een autonome draagrobot ontwikkeld. De L298N motor driver module wordt gebruikt om twee DC-motoren aan te sturen, waardoor de robot in staat is om vooruit, achteruit, links en rechts te bewegen. Deze module biedt niet alleen richting- en snelheidsregeling, maar ook bescherming tegen overbelasting. Dat is gedaan voor het verlengen van het levensduur van de motor, maar ook van de motorprestaties en de duurzaamheid van de robot. Door gebruik te maken van de juiste pinaansturingen kan de robot nauwkeurig en betrouwbaar reageren op commando's van de sensoren, zoals de ultrasone sensor voor obstakeldetectie. De L298N motor driver module is geselecteerd vanwege zijn betrouwbaarheid, efficiëntie en

compatibiliteit met de gebruikte ESP32 microcontroller.

- In de laatste literatuuronderzoek [8] is een EPAC-project (*Extreme Photonics Application Centre*) opgezet van het *Central Laser Facility*. Bij EPAC wordt gebruik gemaakt van een uitgebreide gebruikersinterface voor laseroperators ontwikkeld met het Blazor-framework. Blazor maakt toegang via een webapplicatie en -server mogelijk voor het *Control System Studio*, dat leidt tot een verbeterde flexibiliteit en gebruiksgemak van *end-users* via een *access-point*. EPICS kan worden opgezet op een virtuele machine om lange configuraties te vermijden of het kans zelfs geïntegreerd worden in een *mockup*-fysiek controlesysteem gebouwd met *Raspberry Pi's*. Dit biedt een toegankelijke en kosten-effectieve manier om ervaring op te doen met het platform.
- Deze zes artikelen geven een aantal oplossingsrichtingen die mogelijk van toepassing kunnen zijn voor dit onderzoeksartikel, namelijk:
 - Het gebruik maken van een MySQL database, omdat MySQL een secure dataopslag aanbiedt en complexere datastructuren aankan;
 - Mogelijke toepassing van Blazor voor het ontwikkelen van webapplicaties door middel van C#, wat handig kan zijn voor een betere integratie tussen frontend en backend;
 - Mogelijke integratie van diverse sensoren, zoals: LiDAR, infrarode en ultrasone sensoren voor nauwkeurige metingen in verschillende omgevingen;
 - Het gebruik van een soortgelijke L298N *motordriver*-module voor motorcontrole bij autonome robots;
 - Het mogelijke gebruik van een *Raspberry Pi* om een *access-point* op te zetten en als mastercontroller te gebruiken.

C. Vraagstelling

De hoofdvraag is:

“Met welke hard- en software configuratie is een minimaal werkend systeem te ontwikkelen waarmee de contouren van een ruimte verkend en gevisualiseerd kan worden in een object-georiënteerd webapplicatie?”

Om deze hoofdvraag te kunnen beantwoorden, zijn er vier deelvragen bedacht, namelijk:

- A. Welke sensoren en componenten zijn er minimaal nodig om een werkend systeem op te zetten waarmee een ruimte verkend kan worden om de coördinaten van de navigatie te kunnen bepalen?
- B. Welke optionele sensoren en/of componenten kunnen in een tweede release worden toegevoegd aan het minimaal

werkend systeem om de besturing en output van de hardware accurater te maken?

- C. Op welke manier kunnen de coördinaten van het minimaal werkend systeem object-georiënteerd met een webserver continu gecommuniceerd en verwerkt worden?
- D. Met welke programmeermogelijkheden zijn er een object-georiënteerd webapplicatie te realiseren zodat een verkende ruimte gevisualiseerd kan worden op basis van continue gecommuniceerde en opgeslagen data op de webserver?

Deze deelvragen zijn gericht op het verkrijgen van inzicht in de technische mogelijkheden van de te gebruiken soft- en hardware en de mogelijke oplossingsrichtingen voor het stap voor stap ontwikkelen van een object-georiënteerd en minimaal werkend systeem. Het beantwoorden van deze deelvragen zal uiteindelijk leiden tot een eerste minimaal werkende oplossing voor het visualiseren van de contouren van een ruimte op een webapplicatie en de communicatie tussen de robot en webserver.

III. METHODEN

Voor dit artikel zijn de volgende onderdelen van methode toegepast: formele analyse, ontwerp en experiment.

A. Welke sensoren en componenten zijn er minimaal nodig om een werkend systeem op te zetten waarmee een ruimte verkend kan worden om de coördinaten van de navigatie te kunnen bepalen?

Om deze deelvraag te beantwoorden wordt een literatuuren technische-specificatie onderzoek gedaan. Deze methode is een vorm van deskresearch en het ontwerpen van schakeling(en).

Er is hiervoor gekozen zodat met *reverse-engineering* een bestaande robot toegepast kan worden voor het realiseren van de totale opdracht. Door het gebruiken van een bestaande *Samsung*-robot zal de hardware deels bestaan uit de bestaande hardware van *Samsung* zelf en deels uit nieuwe componenten. Bij hergebruik van *Samsung* componenten zijn de door de fabrikant gehanteerde specificaties leidend om te kunnen gebruiken in het eigen ontwerp.

Literatuuronderzoek en bestuderen van technische specificaties zijn belangrijk om de benodigde informatie te verzamelen voor het opzetten van een onderwerp van een prototype. Ook is hierbij kennis van eerdere gedane onderzoeken handig om paraat te hebben, zodat de gestelde deelvraag op een juiste manier beantwoord kan worden. De te doorlopen stappen hierbij zijn:

- Bestuderen van technische specificaties hoe de benodigde hardware op de juiste manier toegepast kunnen worden:
 - 1) Infrarode sensoren (fabrikant - *Samsung*);
 - 2) Microswitch bumpers (fabrikant - *Samsung*);
 - 3) Motorcontroller;

- Dualchannel Bridge H 12 Volt (DBH-12V).

4) Master-device:

- Raspberry Pi 3b+.

5) Voeding:

- 12V batterij voor motoren;
 - 5V batterij voor *master-device*.

➤ Literatuuronderzoek gericht op het uitlezen van gegevens door een *master-device*.

Voor het ontwerp van een systeem zijn de technische specificaties essentieel om een werkend ontwerp te kunnen realiseren. Daarnaast is het belangrijk om van andere vergelijkbare ontwerpen, die gepubliceerd zijn, te leren en daarop verder door te ontwikkelen.

Als resultaat voor een *Proof Of Concept* (POC) is het van belang om een minimaal werkend systeem op te leveren, waarbij de data van de sensoren uitgelezen, verzameld en verwerkt kunnen worden door een *master-device*. Daarnaast dienen de motoren van de robot op een correcte wijze aangestuurd te worden, door de gekozen motorcontroller, zodat de robot de contouren van een ruimte kan volgen.

B. Welke optionele sensoren en/of componenten kunnen in een tweede release worden toegevoegd aan het minimaal werkend systeem om de besturing en output van de hardware accurater te maken?

Om deze tweede deelvraag te beantwoorden wordt uitgezocht welke eerste mogelijke uitbreidingen met sensoren de gevraagde verbetering mogelijk gemaakt kan worden in een volgende release van het ontwerp. Hiervoor zal gebruik worden gemaakt van een *deskresearch*.

De te doorlopen stappen hierbij zijn:

- Bestuderen van beschikbare artikelen die sensoren en componenten behandelen, die eventueel toegepast kunnen worden in het kader van dit onderzoeksartikel;
- Vergelijk de bijdrage van de gevonden sensoren en componenten de mogelijke potentie in verbetering van de besturing en de output in een volgende release.

Met deze werkwijze kan bestudeerd worden wat de potentie is van het toevoegen van sensoren en/of componenten aan de MVP. Door de verbeterpotentie van de sensoren en/of componenten met elkaar te vergelijken, is het mogelijk om beter te beslissen welke verbetering meer waarde toevoegt aan het onderzoeksproject.

Er is pas voldoende resultaten geboekt wanneer de *deskresearch* een lijst van sensoren en componenten oplevert, waarmee prestaties van de POC mogelijk verbeterd kunnen worden. Dit is een vergelijking van de verwachte bijdrage van extra componenten en/of sensoren in de

opgezette systeem ter verbetering van de prestaties van de robot. De verwachting is dat het met name zal gaan om de nauwkeurigheidsverbeteringen bij de vaststelling van de kaart van een onbekende ruimte en verbeteren van navigeren van de robot door de onbekende ruimte.

C. Op welke manier kunnen de coördinaten van het minimaal werkend systeem object-georiënteerd met een webserver continu gecommuniceerd en verwerkt worden?

Om de derde deelvraag te kunnen beantwoorden is een literatuur- en documentatieonderzoek gedaan.

Voordeel van deze methode is het verkennen van meerdere dataverwerkingsmogelijkheden en gebruik kunnen maken van al eerder uitgevoerde onderzoeken en toegepaste oplossingen. Het is essentieel om een juiste communicatie mogelijkheid te kiezen waarmee de vastgestelde coördinaten op een correcte wijze omgezet kunnen worden tot de contouren van een ruimte. De volgende stappen worden ondernomen om deze deelvraag te kunnen beantwoorden:

- Literatuur- en documentatieonderzoek naar programmeermogelijkheden die gebruikt kunnen worden om via een webserver data te verwerken en te communiceren via een mogelijke database;
- Op welke wijze kan de gekozen programmeermogelijkheid geïmplementeerd worden en welke functies of methoden horen daar bij?

Een literatuur- en documentatieonderzoek leidt tot een gestructureerde inzicht in bestaande technologieën waaruit een passende technologie voor dit project gekozen kan worden. Dit zal ervoor zorgen dat de kans op inefficiënte of onbetrouwbare oplossingen in de te maken keuze aan de voorkant van het project worden uitgefilterd.

Het resultaat is pas succesvol te noemen wanneer het literatuur- en documentatieonderzoek een onderbouwde aanbeveling oplevert voor een efficiënte methode waarmee de coördinaten van de robot continue te kunnen communiceren en verwerken via een webserver middels een database.

D. Met welke programmeermogelijkheden zijn er een object-georiënteerd webapplicatie te realiseren zodat een verkende ruimte gevisualiseerd kan worden op basis van continue gecommuniceerde en opgeslagen data op de webserver?

Om de vierde deelvraag te kunnen beantwoorden is een literatuur- en documentatieonderzoek gedaan.

Literatuur- en documentatieonderzoek is gekozen, omdat het een overzicht geeft van de methoden die andere onderzoekers eerder hebben toegepast voor het ontwikkelen van een object-georiënteerde webapplicatie. De genomen stappen om deze deelvraag te kunnen beantwoorden zijn:

- Literatuur- en documentatieonderzoek uitvoeren naar programmeermogelijkheden om een kaart te kunnen visualiseren?;

- Op welke wijze kan de gekozen programmeermogelijkheid geïmplementeerd worden en welke functies of methoden horen daar bij?;

Deze aanpak geeft inzicht in een analyse tussen verschillende programmeermogelijkheden en het kiezen van de best passende mogelijkheid in dit POC van het project.

Het resultaat is geslaagd als er een valide oplossing gevonden kan worden om op een object-georiënteerde programmeerwijze de coördinatiegegevens gevisualiseerd kan worden.

IV. RESULTATEN

In dit hoofdstuk staan de resultaten van het onderzoek per deelvraag uitgewerkt.

A. Welke sensoren en componenten zijn er minimaal nodig om een werkend systeem op te zetten waarmee een ruimte verkend kan worden om de coördinaten van de navigatie te kunnen bepalen?

Om deze deelvraag te kunnen beantwoorden is een deels defecte *Samsung* robotstofzuiger aangeschaft, die eerst uit elkaar is gehaald om te zien welke sensoren in dat systeem zijn toegepast. Deze versie van de robotstofzuiger was gebaseerd op twee bumpersensoren en zeven IR-sensoren aan de voorkant, drie IR-sensoren aan de onderkant, twee wiel druksensoren, één camera en besturingshardware en software. De hypothese die wij als onderzoekers hebben gesteld is: “*het moet mogelijk zijn om met minimaal de twee bumpersensoren en eventueel met één IR-sensor aan de voorkant van de robot autonoom kan navigeren om de contouren van een ruimte te volgen*”. Voor het bewijzen van de gestelde hypothese is de onderzoeksmethode “experiment” toegepast. Om te kunnen experimenteren is er eerst een basis ontwerp nodig voor de op te zetten besturing van de robot en kennis van de toe te passen sensoren en componenten. Voor de kennis van de sensoren en componenten zijn de onderstaande deskresearch onderzoeken uitgevoerd.

Nota Bene: De *datasheets* van *Samsung* eigen sensoren zijn niet publiekelijk en is om die reden gezocht naar vergelijkbare sensor *datasheets*.

- Bestuderen van technische specificaties hoe de benodigde hardware op de juiste manier toegepast kunnen worden

Het bleek dat de werking van “*Arduino Infrared Obstacle Avoidance Sensor Module*” overeen te komen met de werking van *Samsung* infrarode sensoren. Die *datasheets* hebben we als basis gebruikt om de technische specificaties als basis te kunnen realiseren. Om dit te bevestigen hebben we geëxperimenteerd met een van de sensoren en de meetresultaten vergeleken met elkaar en dat kwam overeen. De aangenomen hypothese is hiermee bevestigd.

Volgens de [datasheet](#) [9] van de infrarode sensor zijn de onderstaande gegevens belangrijk als uitgangspunt.

De *Samsung* infrarode sensor heeft in totaal vier pennen, namelijk:

- Pen-1: De V_{cc} ingang met een spanningsingang van 3 tot 5 Volt DC (*Direct Current*);
- Pen-2: De data pen. Deze kan worden aangesloten aan een van de GPIO (*General Purpose Input Output*) pennen van een device-controller;
- Pen-3: De *GND* pen. Dit is de grond-pen;
- Pen-4: De *F*-pen. Kan gebruikt worden voor het instellen van een specifieke frequentie waarmee de LED knippert.

Het enige verschil is dat de *Samsung* infrarode sensor een extra *F*-pen heeft. Tijdens dit onderzoek is er gekozen om deze pen aan te sluiten aan 3V DC, omdat er geen behoefte is om een bepaalde frequentie te gebruiken.

De *microswitch* van de *Samsung* robot is helaas ook niet beschikbaar, maar gelukkig zijn microswitches makkelijk qua werking. Uit onderzoek is gevonden dat bijvoorbeeld deze volgende *microswitch* een zelfde werking heeft als die van de *Samsung* robot [10]. Het enige verschil is dat deze datasheet microswitches betreft die functioneren op andere spanningsniveau's.

Vanuit het experiment is uitgegaan van [deze gegevens](#). De *Samsung* microswitch sensor heeft in totaal drie pennen, namelijk:

- Pen-1: De V_{cc} ingang met een spanningsingang van 3 Volt DC;
- Pen-3: De *NO* pen. Deze is aangesloten aan de GPIO van de *device-controller*;
- Pen-3: De *NC* pen en wordt niet gebruikt;

De *microswitch* werkt net zoals elke andere *microswitch* en het aansluiten is geen probleem. De *motorcontroller* die ingezet wordt is de DBH-12V, dit staat voor "Dual-bridge-H-12-Volt". Zie [hier](#) een lijst van technische specificaties van de DBH-12V [11]:

De DBH-12V *motorcontroller* heeft in totaal negen pennen, namelijk [12]:

- Motor V+: De V_{cc} ingang met een spanningsingang van 3 tot 5 Volt DC;
- GND: De data pen. Deze kan worden aangesloten aan een van de GPIO (*General Purpose Input Output*) pennen van een *device-controller*;
- IN1 (B, A): De *IN1*-pen, dit is eerste input pen voor PWM;
- IN2 (B, A): De *IN2*-pen, dit is tweede input pen voor PWM;
- EN (B, A): De *Enable*-pen, dit is de enable pen, voor timing, aan/uit e.d;

- CT (B, A): De *Current*-pen, dit is de current monitor pen;
- A met twee ingangen voor plus en min (motor-kant);
- B met twee ingangen voor plus en min (motor-kant);
- PWR (power) ingang met plus en min voor de motoren (motor-kant).

Voor het aansluiten van deze *motorcontroller* moet goed opgelet worden dat pennen niet door elkaar gehaald worden, want verkeerde aansluitingen kunnen lijden tot een einde van de Raspberry Pi. Vanuit de Raspberry Pi worden twee PWM (Pulse Width Modulation)-pennen per motor (A, B) aangesloten aan beide PWM ingangen van de motorcontroller (dit is voor het moduleren van de snelheid van de motoren). Plus en min van de A-zijde worden aangesloten voor de eerste motor. Plus en min van de B-zijde worden aangesloten voor de tweede motor. Tenslotte wordt de PWR aangesloten aan de 12V accu. Er is gekozen voor deze DBH-12V *motorcontroller*, omdat de L298N uit vooronderzoek niet geschikt was in verband met spannings-eisen van de gebruikte motoren. De gekozen device-controller (master) is een Raspberry Pi 3b+, omdat dit een snelle en duurzame mini computer is. Zie een [hier](#) lijst van algemene technische specificaties van de Raspberry Pi 3b+ [13]:

De gekozen (motor) accu is de 12 Volt accu pack van *ledstrip-koning*, omdat deze een hoge capaciteit en genoeg spanning levert voor de motoren (willen een spanning van 12V). Zie [hier](#) een lijst van algemene technische specificaties van de 12 Volt accu pack [14]:

De gekozen (Raspberry Pi) batterij is de 5V "Lithium Polymer Power Bank" van *rspro*, omdat deze klein van formaat is en genoeg spanning levert voor de Raspberry Pi. Zie [hier](#) een lijst van algemene technische specificaties van de 5 Volt RS-PRO powerbank [15]: Voor het systeemontwerp zie [hier het blokschema](#) van de hardware als resultaat.

Met de verkregen inzichten over sensoren en componenten is er een minimale werkende hardware ontworpen zoals weergegeven in het bovengenoemde blokschema. Waarna geëxperimenteerd is om de robot autonoom te laten navigeren door een ruimte om de contouren van de ruimte te volgen. Tijdens experimenteren vielen een aantal zaken op, zoals met de bumpersensoren viel op als de robot tegen een muur aan reed deze ingedrukt bleef of de robot ging draaien bij een bocht omdat de motor dan niet stopt. Door de toevoeging van kill switch in het ontwerp kon er tijdens de test fase de autonome robot gestopt worden als er iets fout ging tijdens navigeren. Dat heeft bijgedragen als een soort "*debug*" functie. Ook is er een probleem ontstaan met gebruik van IR-sensor bij donkere kleuren van de muren, deuren en dergelijke. Dan werken de IR-sensoren namelijk niet goed doordat IR-signaal geabsorbeerd wordt door de donkere oppervlakte (geen terugkaatsing van het signaal). Deze problemen met de navigatie zijn opgelost in de besturingssoftware. Voor de

bumpersensor is er een timer/stopwatch gebruikt waarmee gedetecteerd kan worden of de robot vast is gelopen. Indien het geval dan wordt de robot een stuk naar achteren gereden en daarna met een curve naar voren gereden. Dit zorgt voor dat de robot slingerend de contouren volgt. Het draaien is opgelost door in de software het deel uit te halen dat van uitging als er geen muur detectie was ga dan naar rechts om een muur te vinden. Het probleem van de IR-sensor bij donkere kleuren is in de POC versie opgelost door de bumper-input op die momenten het rijgedrag te laten bepalen.

De resultaten van de POC versie is redelijk, maar kent nog bepaalde nadelen met de infrarood sensor in combinatie met de 2 bumpers wat zich uit als minder accurate output en stabielere navigatie van de robot.

- Literatuuronderzoek gericht op het uitlezen van gegevens door een *master-device*

Uit het literatuuronderzoek gericht op het uitlezen van gegevens door een *master-device* is er geen officiële literatuur gevonden dan de officiële handleiding waarmee het uitlezen van gegevens mogelijk is geworden. Hiervoor is de handleiding “*pigpio*” gebruikt [16]. Het resultaat is toepassen van onder andere de onderstaande methoden:

- Essential:
 - gpioInitialise: initialiseert de bibliotheek;
 - gpioTerminate: Stopt de bibliotheek.
- Basic:
 - gpioSetMode: Stel GPIO modus in;
 - gpioGetMode: Verkrijg GPIO modus;
 - gpioSetPullUpDown: Set/clear GPIO pull up/down weerstand;
 - gpioRead: Leest de GPIO-pen uit;
 - gpioWrite: Schrijft naar de GPIO-pen.

B. Welke optionele sensoren en/of componenten kunnen in een tweede release worden toegevoegd aan het minimaal werkend systeem om de besturing en output van de hardware accurater te maken?

- Bestuderen van beschikbare artikelen die sensoren en componenten behandelen, die eventueel toegepast kunnen worden in het kader van dit onderzoeksartikel

Uit literatuuronderzoek blijkt dat er meerde sensoren toegepast kunnen worden om de besturing en output van de autonome robot te kunnen verbeteren met onder andere:

- een magnetometer meet het magnetische veld rond de robot die als kompas kan dienen om draaiingen nauwkeuriger te detecteren. Echter is deze sensor ook kwetsbaarheid voor magnetische storingen in een omgeving [17], [18];

- een gyroscoop meet het momentane impulsmoment rond elke as. Hiermee zou ook de draaiing nauwkeuriger gedetecteerd kunnen worden [17], [18];
- een microcontroller om de rotary encoder module uitlezen van motoren ter ondersteuning van het bepalen van aantal omwentelingen van de motoren. Dit kan bijvoorbeeld helpen om een *homebase* van de robot terug te vinden [19];
- een LiDAR sensor die bepaalt de *traject tracking* met behulp van een laserafstand-meter. Dit kan ondersteunen bij de positie en oriëntatie bepaling van de robot in te schatten [20].
- Vergelijk de bijdrage van de gevonden sensoren en componenten de mogelijke potentie in verbetering van de besturing en de output in een volgende release

In het vorige kopje “*Bestuderen van beschikbare artikelen die sensoren en componenten behandelen, die eventueel toegepast kunnen worden in het kader van dit onderzoeksartikel*” is onderzocht naar eventuele sensoren die toegepast kunnen worden. Het is belangrijk om de verschillen tussen deze sensoren en hun potentie te kennen om vervolgens de juiste keuze te maken voor een mogelijke tweede release. Uit de artikelen bleek onder andere dat de magnetometer in principe hetzelfde werkt als een kompas. De magnetometer gebruikt magnetische veld van de aarde, waarmee bepaald kan worden naar welke pool het object georiënteerd staat. Door deze sensor toe te passen in een volgende release kan er op een effectievere wijze bepaald worden wanneer de robot een draaiing maakt. Het enige nadeel van deze sensor betreft magnetische storingen van buitenaf, zoals verstoringen door vermogenslektronica of magnetische objecten.

De gyroscoop werkt net iets anders dan de magnetometer, het meet het momentane impulsmoment rond elke as en kan draaiingen makkelijk bepalen om elke as. Gyroscopen kunnen ook worden ingezet als kompas en zijn deze nauwkeuriger dan magnetometers, omdat de gyriscoop niet zo makkelijk verstoord wordt als de magnetometer. Het nadeel van de gyroscoop is dat deze om de zoveel tijd wel gekalibreerd moet worden omdat de nauwkeurigheid ervan verloopt.

Uit het literatuuronderzoek is gebleken dat een combinatie van de gyroscoop en magnetometer betere resultaten kan opleveren dan deze twee individueel toepassen.

Ook kan een LiDAR toegepast worden, deze sensor kwam eerder terug in het vooronderzoek. In dat vooronderzoek is gekeken naar welke soorten sensoren toegepast kunnen worden voor het meten van hoogte bij onbemande luchtvaartuigen [6]. Uit dat onderzoek bleek onder andere dat LiDAR’s een hoge nauwkeurigheid hebben en kunnen helpen bij het efficiënter maken van draaiingen en object herkenningen [19].

Voor de eerste release is er een [hardware blokschema ontworpen](#). Hierin is een *sensorhub*-blok te zien, waar alle sensoren bij elkaar komen, in een sensor HUB. Er is gekozen om op de

sensorhub twee *sockets* beschikbaar te stellen waarop twee microcontrollers aangesloten kunnen worden om de rotary encoder uit te kunnen lezen. Het ontworpen CAD-schema van de *sensorhub* kan [hier](#) gevonden worden. Ook is er rekening gehouden om een magnetometer toe te passen voor in een volgende release en is deze sensor meegenomen in het blokschema en aanwezig in de hardware.

De beschouwde sensoren in de deskresearch hebben voor en nadelen om het rijgedrag van de robot te kunnen stabiliseren en de resultaten te verbeteren. Als de sensoren met elkaar worden vergelen welke de beste potentie heeft om in de tweede release toe te voegen dan valt het volgende op: De magnetometer en gyroscoop hebben naast voordelen ook belangrijke nadelen die beheerst moeten worden. De verwachting is dat toepassing van de combinatie van deze twee sensoren complexer zal zijn dan toevoegen van een LiDAR sensor. De uitbreiding met alleen magnetometer, die al reeds aanwezig is in de hardware, heeft teveel nadelen om te beheersen. De toegevoegde waarde van de microcontrollers om de rotary encoder uit te kunnen lezen is beperkt t.o.v. andere aanwezige standaard sensoren zoals de overige zes IR-sensoren aan de voorkant.

Afwegend tussen benodigde moeite en verbeterpotentie is de LiDAR sensor en/of de zes IR-sensoren aan te bevelen om toe te voegen in de tweede release aan het minimaal werkend systeem om de besturing en output van de hardware accurater te maken.

C. Op welke manier kunnen de coördinaten van het minimaal werkend systeem object-georiënteerd met een webserver continu gecommuniceerd en verwerkt worden?

- Literatuur- en documentatieonderzoek naar programmeermogelijkheden die gebruikt kunnen worden om via een webserver data te verwerken en te communiceren via een mogelijke database

In de [eerste](#) en [tweede](#) vooronderzoek kwam terug dat er twee soorten database zijn, namelijk relationeel (MySQL) en non-relationeel (NoSQL). De belangrijkste punten waren dat MySQL veel consistentier met complexere queries te zijn en is MySQL veel secuurder te zijn ten op zichte van NoSQL [3], [4]. Omdat MySQL beter presteert bij leesbewerkingen dan NoSQL, is op basis van vooronderzoeken gekozen om MySQL toe te passen in dit onderzoeksartikel. De database-server moet worden toegepast op Raspberry Pi dat draait op operating-systeem Linux. Linux biedt meerdere soorten MySQL-server en client-applicaties en wordt de volgende server aangeraden, namelijk: “MariaDB” [21].

Als er een database wordt opgezet dan moet de communicatie (webinterface) nog eerst worden opgebouwd met een object-georiënteerd programma.

Om op een nette manier data te communiceren tussen twee programma's is het nodig om een API (*Application Programming Interface*) te implementeren in het onderzoekresultaat. Een API is de *backend* van een webapplicatie en is de

enige applicatie die rechtstreeks een connectie heeft met de database. Met een API is het vervolgens mogelijk om CRUD (*Create, Read, Update, Delete*) commando's te implementeren die aangeroepen kunnen worden door elke andere programma door middel van een “HTTP-request”. Het voordeel van een API is dat communicatie tussen applicaties en de database veel simpler, mooier en duidelijker maakt in plaats van meerdere directe connecties te hebben naar een database. Bovendien gebruiken de meeste API's een object-georiënteerde (abstracte) manier om data te verwerken en dat is: de JSON (*JavaScript Object Notation*)-formaat [22].

In C# is het mogelijk om een connectie te leggen met de MySQL database om te kunnen data te kunnen communiceren [23]. Naast deze voordelen kent C# ook IoT-support [24]. Daarnaast is er nog een alternatief en dat is: *Java Web Development*. Binnen programmeertaal Java is het ook mogelijk om op een object-georiënteerde manier te communiceren met de database. Daarnaast biedt Java ook de mogelijkheid om een API op te zetten net als in C# die CRUD commando's aanwezig stelt [25].

In de [derde](#) en [zesde](#) vooronderzoek kwam ook terug dat onderzoekers het Blazor-framework gebruiken om in C# zowel een webserver als webapplicatie op te zetten. Dit hadden deze onderzoekers gedaan, omdat het Blazor-framework een versimpelde en object georiënteerde integratie van zowel frontend als backend aanbiedt. In dit onderzoek is gekozen om C# als programmeermogelijkheid toe te passen, omdat C# meer IoT-support aanbiedt ten opzichte van Java [24], [26], [27]

- Op welke wijze kan de gekozen programmeermogelijkheid geïmplementeerd worden en welke functies of methoden horen daar bij?

Hieronder staat een lijst van basis commando's om MariaDB te kunnen gebruiken [21]:

- Verbinden naar database: `mysql <database-naam>`
- Help: `help contents`
- Lijst databases: `SHOW DATABASES;`
- Verander atabase: `USE <database-naam>`
- Lijst tabellen: `SHOW TABLES;`
- Maak DB: `CREATE DATABASE <database-naam>;`
- Verwijder DB: `DROP DATABASE <database-naam>;`
- Tabel info: `DESCRIBE <database-naam>;`
- Laad data: `LOAD DATA INFILE '<bestand-naam>'`
- Verlaat client: `QUIT` of `EXIT`
- Inloggen met root: `mysql -u root -p`

Een databaseconnectie opzetten in C# gaat als volgt:

- initialiseer en declareer connection-string;

- maak “SqlConnection”-object en geef connection-string mee.

Wel geeft het boek aan dat het C# bibliotheek *System.Data.SqlClient* vereist is [23].

Een API in C# bestaat uit de volgende onderdelen [28]:

- De significante term “MVC” staat voor: Model, View en Controller;
- Een “Model” bevat alle data/componenten die in een database horen en daar hetzelfde genoemd zijn;
- Een “View” is de *User Interface* oftewel de webapplicatie/frontend;
- Een “Controller” is hetgene wat de gebruikersinteractie afhandelt en het model en database *up-to-date* houdt (backend).

Om een webserver te implementeren op een object-georiënteerde wijze moet er eerst een klassediagram ontworpen worden, waarbij het MVC structuur terug te zien moet zijn. Aanvankelijk heeft de auteur van dit onderzoeksartikel een [object georiënteerde klassediagram concept](#) ontworpen. Dit met het idee dat het nog geschreven programmeercode moet slaan op het object georiënteerde klassediagram. Echter is het klassediagram anders eruit gaan zien dan aanvankelijk bedacht was. Dit omdat MVC-principe slecht terug te zien was in het aanvankelijke klassediagram en het structuur niet compleet was voor zowel de webserver als webapplicatie. Daarom is door de auteur van dit onderzoeksartikel gekozen om dit klassediagram een [vernieuwde versie](#) te ontwerpen en toe te passen. Het verschil tussen de [vernieuwde](#) en [aanvankelijke](#) klassediagrammen is erg significant, namelijk dat in het vernieuwde:

- Klassediagram duidelijker het MVC-principe met de CRUD commando’s terug laat zien;
- Dat nu een lijst gebruikt wordt voor de coördinatiegegevens middels een (*primary key*) ID voor de database;
- Dat de aanvankelijke “SensorModel” nu veranderd is naar “CoordModel”, wat staat voor het Coördinaat-Model;
- Dat de geïmplementeerde API toch groter is dan gedacht.

Een mooie resultaat en goed om terug te zien dat het “Model” in beide UML-klassediagrammen terug te zien is (wat onderdeel is van het MVC-principe).

D. Verken programmeermogelijkheden voor het opzetten van een object-georiënteerd webapplicatie dat een ruimte kan visualiseren aan de hand van opgeslagen coördinaten in de webserver?

- Literatuur- en documentatieonderzoek uitvoeren naar programmeermogelijkheden om een kaart te kunnen visualiseren?

Dotnet, het open-source C# framework van Microsoft biedt WASM (Web ASseMbly) aan, waarmee zowel een webapplicatie als webserver opgezet kan worden in programmeertaal C# [29]. Dotnet heeft de volgende voordelen:

- Het open-source dotnet gebruikt, wat ook op de Raspberry Pi toegepast kan worden;
- Het project gebouwd kan worden voor ARM (*Acorn RISC Machine*)-architectuur op een andere host PC (*Personal Computer*) [30];
- Het ondersteunt makkelijk (cross-)debuggen van programma’s om fouten snel op te sporen [31].

Om een kaart te kunnen visualiseren in C# kan er gebruik gemaakt worden van een HTML5 canvas [32]. Ook bestaat er een uitbreiding van de HTML5 canvas, namelijk Skia. Voor Web Assembly bestaat er ook een versie van Skia, namelijk SkiaSharp, gecompileerd vanuit C++ source met WASM compiler [33], [34].

Een andere optie is Java Web Application, hiermee is het mogelijk om een webapplicatie te bouwen op een object-georiënteerde wijze binnen programmeertaal Java. Voordelen van Java zijn:

- Het is open-source framework dat makkelijk op de Raspberry Pi toegepast kan worden;
- Het project kan gebouwd worden op de Raspberry Pi zelf [35].

Om een kaart te kunnen visualiseren in een Java Web Application kan er gebruik gemaakt worden van een HTML5 canvas of een JavaScript-canvas toe te passen [36].

Uit vergelijking is er gekozen om C# te gebruiken voor dit onderzoeksartikel, omdat er voor de webserver ook gekozen is voor programmeermogelijkheid C# in combinatie met het Blazor-framework. Bovendien is debuggen in programmeertaal C# makkelijker ten op zichte van Java.

- Op welke wijze kan de gekozen programmeermogelijkheid geïmplementeerd worden en welke functies of methoden horen daar bij?

Om op een object-georiënteerde manier een kaart te visualiseren wordt er gebruik gemaakt van C#-Blazor in combinatie met SkiaSharp. SkiaSharp biedt de volgende methodes aan om coördinaten te kunnen tekenen binnen een rechthoek op een canvas [37]:

- SkCanvas() creëert een lege canvas, maar je kan ook parameter meegeven, zoals de breedte, hoogte, kleur, etc;
- SkRect(breedte, hoogte, canvas-oppervlak-breedte, canvas-oppervlak-hoogte) creëert een rechthoek;
- SkCanvas.Clear() verwijdert alles uit de canvas;
- SkPaint(stijl, kleur, etc): hiermee kun je een eigen stijl *preset* maken;

- `SkCanvas.DrawCircle(X, Y, straal, SkPaint)`: hiermee maak je een cirkel/punt aan;
- `SkCanvas.DrawRect(SkRect, SkPaint)`: hiermee maak je een cirkel/punt aan [37].

V. CONCLUSIES

A. Vraagstelling

De hoofdvraag “*Met welke hard- en software configuratie is een minimaal werkend systeem te ontwikkelen waarmee de contouren van een ruimte verkend en gevisualiseerd kan worden in een object-georiënteerd webapplicatie?*” is opgedeeld in vier deelvragen voor een gestructureerde aanpak en uitwerking van het onderzoek. Dit aanpak heeft geholpen om succesvol te kunnen werken naar de beoogde resultaat van het project.

B. Welke sensoren en componenten zijn er minimaal nodig om een werkend systeem op te zetten waarmee een ruimte verkend kan worden om de coördinaten van de navigatie te kunnen bepalen?

Deze deelvraag omvat een hardware opzet en communicatie software (webinterface) om zo op een efficiënte manier de coördinatiegegevens van de autonome robot continue vast te stellen en te communiceren met de webserver. De toegepaste onderzoeksmethode voor deze deelvraag en de doorlopen sub-stappen hebben geleid tot een succesvolle resultaat. De coördinatiegegevens van de autonome robot worden continue vastgesteld en verwerkt via de webserver met de ontworpen hard- en software. De minimale hardware hiervoor bestaat uit twee bumper sensoren en één IR sensor voor het autonoom laten navigeren van de robot. Door de toevoeging van *kill-switch* in het ontwerp kon er tijdens de test fase de autonome robot gestopt worden als er iets fout ging met navigeren. Dat heeft bijgedragen als “*debug*” functie. De resultaten van de POC versie is redelijk, maar kent nog bepaalde nadelen met de infrarood sensor in combinatie met de twee bumpers wat zich uit als minder accurate output en stabielere navigatie van de robot.

C. Welke optionele sensoren en/of componenten kunnen in een tweede release worden toegevoegd aan het minimaal werkend systeem om de besturing en output van de hardware accurater te maken?

Uit de deskresearch van mogelijke toe te voegen sensoren en componenten zijn onder andere de volgende mogelijkheden gevonden om het rijgedrag van de robot te kunnen stabiliseren en de resultaten te verbeteren:

- magnetometer;
- gyroscoop;
- LiDAR sensor;
- zes aanwezige IR-sensoren *Samsung*;
- toevoeging twee microcontrollers ten behoeve van rotary encoder.

Uit de vergelijking van de verbeter potentie en de moeite die nodig is om werkend te krijgen is de magnetometer, gyroscoop en de toevoeging twee microcontrollers afgevalen daar deze of belangrijke nadelen hebben die beheerst moeten worden of weinig toegevoegde waarde kennen bij het oplossen van het gestelde probleem of de complexiteit. Geconcludeerd wordt dat de verbeterpotentie van de LiDAR sensor en/of de zes IR-sensoren de moeite waard is om toe te voegen in de tweede release aan het minimaal werkend systeem om de besturing en output van de hardware accurater te maken.

D. Op welke manier kunnen de coördinaten van het minimaal werkend systeem object-georiënteerd met een webserver continu gecommuniceerd en verwerkt worden?

De literatuur en documentatieonderzoek heeft geleid tot goede inzichten in verschillende oplossingsmogelijkheden. Dat heeft geresulteerd in gebruik van API ten behoeve van communicatie en de database opzet op een efficiënte manier. Daarnaast zijn de voor- en nadelen van het toepassen van C# en Java vergeleken om de webserver op te zetten. Er is gekozen om C# als programmeermogelijkheid toe te passen, omdat C# meer IoT-support aanbiedt ten opzichte van Java. Om een webserver te implementeren op een object-georiënteerde wijze is het noodzakelijk om eerst een klassediagram te ontwerpen, waarbij het MVC structuur terug te zien moet zijn.

E. Met welke programmeermogelijkheden zijn er een object-georiënteerd webapplicatie te realiseren zodat een verkende ruimte gevisualiseerd kan worden op basis van continue gecommuniceerde en opgeslagen data op de webserver?

De literatuur en documentatieonderzoek heeft geleid tot voldoende inzichten in oplossingsmogelijkheden met de voor- en nadelen ervan om een kaart te kunnen visualiseren. Uit het onderzoek zijn er twee opties voor programmeermogelijkheden geselecteerd, namelijk in C# en Java Web Application. Beide bieden vele voordelen en mogelijkheden die deels vergelijkbaar zijn om een webapplicatie te realiseren. Echter, uit de gemaakte vergelijking is er voor C# gekozen, omdat er voor de webserver ook C# wordt gebruikt en bovendien is debuggen in programmeertaal C# makkelijker dan ten opzichte van Java. Om op een object-georiënteerde manier een kaart te visualiseren wordt voor de implementatie gebruik gemaakt van C# in combinatie met *SkiaSharp*.

F. Eindconclusie

Alle vier de deelvragen zijn succesvol doorlopen en hebben geleid tot een deel resultaten die tezamen mogelijk hebben gemaakt om de hoofdvraag “*Met welke hard- en software configuratie is een minimaal werkend systeem te ontwikkelen waarmee de contouren van een ruimte verkend en gevisualiseerd kan worden in een object-georiënteerd webapplicatie?*” te beantwoorden. Omdat er een minimaal werkend systeem met dit project opgeleverd is, wat tot een eerste gewenste resultaat heeft geleid, kan er geconcludeerd worden dat de uitvoering van dit onderzoeksproject geslaagd is.

VI. DISCUSSIE

In de discussie wordt gereflecteerd naar de werkwijze van hoe het hard- en software ontwerp heeft geleid tot een minimaal werkend systeem. Hierbij worden de successenpunten van het POC en de mogelijke verbeterpunten van het onderzoek voor een volgende release besproken.

In dit onderzoek zijn de basis sensoren en componenten toegepast om de coördinaten van de contouren van een ruimte te kunnen bepalen. Deze hebben in de POC geleid tot globale contouren, die soms nog onnauwkeurig of accuraat genoeg zijn. De besturing is nu nog gebaseerd op een soort slinger beweging van de robot ten op zichte van een muur, dit is gebaseerd op de bumper detectie. Ook bij een bocht is de navigatie van de robot nog berustend op de bumper-input en het bijsturen van rijgedrag. Verder hebben de infrarode sensoren last van oppervlakte kleuren. Bij donkere kleuren, bijvoorbeeld zwart, wordt het signaal geabsorbeerd en niet teruggekaatst, waardoor de bumper-input het rijgedrag bepaald. Dit heeft te maken van de keuze voor besturingswijze van de autonome besturing van de robot. Dit kan verbeterd worden door toevoeging van andere componenten en sensoren in het opgezette systeem. Zoals onderzocht met deskresearch onder de deelvraag B. Door toevoeging van bijvoorbeeld een LiDAR sensor kan de ruimte en obstakels beter in kaart worden gebracht waardoor de robot beter autonoom door een ruimte kan navigeren. Dat zal resulteren in een betere terugkoppeling van de coördinaten waarmee een kaart van de ruimte wordt gevisualiseerd.

Ook zijn er nog andere sensoren beschouwd in de deskresearch die het rijgedrag van de robot kunnen stabiel maken. Alleen hebben de twee uitgezochte sensoren, magnetometer en gyroscoop, naast voordelen ook nadelen die beheerst moeten worden. De verwachting is dat toepassing van de combinatie van deze twee sensoren complexer zal zijn dan toevoegen van LiDAR sensor, die het resultaat beduidend zal verbeteren ten opzichte van de standaard sensoren en componenten zoals nu toegepast in de POC versie van de robot.

Een mogelijk tweede iets minder complexe verbeterpunt kan zijn het toepassen van al beschikbare sockets op de PCB waarop twee microcontrollers aangebracht kunnen worden waarmee rotary encoder uitgelezen kan worden. Daarnaast wordt momenteel slechts één infrarode sensor gebruikt, terwijl uit de resultaten gebleken is dat de rest van de (6) IR sensoren toepassen betere resultaten zullen opleveren. Ook is er rekening gehouden om een magnetometer toe te passen voor in een volgende release en is deze sensor meegenomen in het blokschema. Of de aanwezige uitbreidingsmogelijkheden, rotary encoder en magnetometer, de benodigde verbetering teweeg kunnen brengen is de vraag. Ten op zichte van LiDAR sensor is de verwachting dat de bijdrage minder zal zijn en de toevoeging complexer.

Een andere kleine verbeterpunt webapplicatie betreft is dat de gevisualiseerde kaart vergroot kan worden, omdat de kaart klein eruit ziet.

De succespunten van de POC en het uitgevoerde onderzoek is een minimaal werkend systeem (MVP) die ook verder ontwikkeld kan worden. Het laat tevens ook een ander potentie zien, wat niet tot onderzoeksvraag behoort, dat er met *reverse engineering* een bestaande robot *refurbished* kan worden en nieuwe functionaliteiten toegevoegd kan worden onafhankelijk van de fabrikant. Dit is in dit onderzoek bewezen door gebruik te maken van een deels defecte *Samsung* robotstofzuiger te gebruiken en deze met eigen ontworpen besturingssysteem aan te passen. In dit onderzoek is zowel hard- als software ontworpen met deels toevoeging van andere componenten. Ook is er software ontwikkeling mogelijk gemaakt in andere omgeving dan de fabrikant dat heeft toegepast.

Het bestuderen van artikelen van andere onderzoekers heeft een breder inzicht gegeven en bijgedragen op het domein van toepassen van sensoriek en ontwikkelen van benodigde software op het gebied van autonoom besturen van een robot, visualiseren van een ruimte en verbeter mogelijkheden van de resultaten van een POC.

VII. AANBEVELINGEN

Hoewel het onderzoek succesvol was in het ontwikkelen van een minimaal werkend systeem (POC), is dit nog maar een eerste stap als resultaat van het onderzoeksvraag. Er zijn nog veel verbeteringen mogelijk in accurate besturing, communicatie van benodigde coördinaten om een kaart van een ruimte te visualiseren en het toevoegen van nieuwe functionaliteiten.

In een vervolgonderzoek is aan te bevelen om als eerste verbeter stap de navigatie prestaties en het in kaart brengen van een ruimte te verbeteren. Dit kan deels door besturingssoftware nog verder te verfijnen en deels door hardware uit te breiden met een LiDAR sensor en/of toevoegen van de zes overige IR-sensoren. Dit is belangrijk om stap voor stap te ontwikkelen en vast te stellen of de behaalde resultaten voldoende zijn om pas daarna weer uit te breiden. Eigenlijk wordt hier aangeven om het project op basis van *agile-principes* uit te voeren.

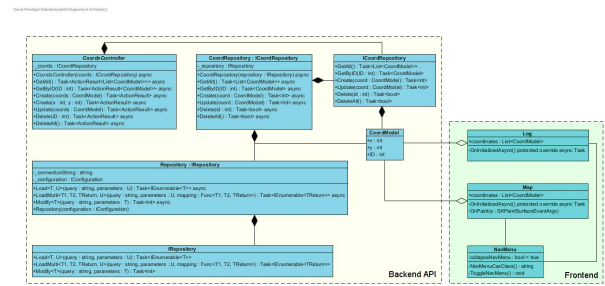
DANKBETUIGINGEN

De auteur wil zijn dank uitspreken aan Sander Gieling. Doordat de heer S. Gieling zijn kennis heeft gedeeld over het formeel schrijven van een onderzoeksartikel tijdens de lessen afgelopen jaar, heeft de auteur van dit artikel een beter inzicht gekregen over het schrijven van een onderzoeksartikel.

REFERENTIES

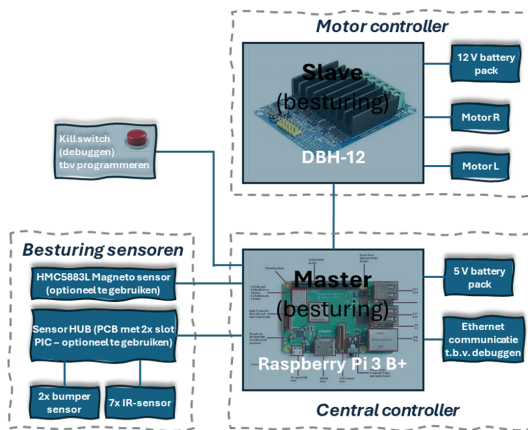
- [1] H. H. Prayash, M. R. Shaharear, M. F. Islam, S. Islam, N. Hossain, and S. Datta, "Designing and optimization of an autonomous vacuum floor cleaning robot," *Robotics Automation Artificial-intelligence*, Nov. 2019. [Online]. Available: <http://dx.doi.org/10.1109/RAAICON48939.2019.11>
- [2] I. Ulrich, F. Mondada, and J.-D. Nicoud, "Autonomous vacuum cleaner," *Robotics and Autonomous Systems*, vol. 19, no. 3–4, p. 233–245, Mar. 1997. [Online]. Available: [http://dx.doi.org/10.1016/S0921-8890\(96\)00053-X](http://dx.doi.org/10.1016/S0921-8890(96)00053-X)
- [3] P. Filip and L. Cegan, "Comparison of mysql and mongodb with focus on performance," *Informatics Multimedia CyberInformation*, Nov. 2020. [Online]. Available: <http://dx.doi.org/10.1109/icimcis51567.2020.9354307>
- [4] S. Rautmare and D. M. Bhalerao, "Mysql and nosql database comparison for iot application," *Informatics Computer Applications*, Oct. 2016. [Online]. Available: <http://dx.doi.org/10.1109/ICACA.2016.7887957>
- [5] I. Clitan, A. Puscasiu, V. Muresan, M. L. Unguresan, and M. Abrudean, "Web application for statistical tracking and predicting the evolution of active cases with the novel coronavirus (sars-cov-2)," *Int. J. Modeling Optim.*, vol. 11, pp. 70–74, 2021. [Online]. Available: <https://www.ijmo.org/vol11/780-CR04.pdf>
- [6] D. M. Randjelovic, G. S. Vorotovic, A. Bengin, and P. N. Petrovic, "Quadcopter altitude estimation using low-cost barometric, infrared, ultrasonic and lidar sensors," *FME Transactions*, vol. 49, no. 1, pp. 21–28, 2021. [Online]. Available: <https://scindeks.ceon.rs/article.aspx?artid=1451-20922101021R>
- [7] N. A. B. A. HALIM, "Wireless autonomous carying robot," 2023. [Online]. Available: http://repository.psa.edu.my/bitstream/123456789/4543/1/1.%20REPORT_NURUL%20ASDAQ%20BIN%20ABDUL%20HALIM.pdf
- [8] S. Feister, K. Cassou, S. Dann, A. Döpp, P. Gauron, A. J. Gonsalves, A. Joglekar, V. Marshall, O. Neveu, H.-P. Schlenvoigt, and et al., "Control systems and data management for high-power laser facilities," *High Power Laser Science and Engineering*, vol. 11, p. e56, 2023. [Online]. Available: <http://doi.org/10.1017/hpl.2023.49>
- [9] *Arduino IR Infrared Obstacle Avoidance Sensor Module*, imimg. [Online]. Available: https://5.imimg.com/data5/YT/KV/MY-1833510/arduino-ir-infrared-obstacle-avoidance-sensor-module.pdf&ved=2ahUKEwjV2r_YkN6GAxVE0gIHHXHbCFYQFnoECBEQAQ&usg=AOvVaw03lxw3_Hvfr8v83Iy3RxX8
- [10] *Micro Switch or Snap-action Switch*, components101. [Online]. Available: <https://components101.com/switches/micro-switch-or-snap-action-switch>
- [11] *DC 512V 0A30A Motor Controller Dual-Channel H Bridge Motor Driver Board Module*, Hilitand. [Online]. Available: <https://www.amazon.ca/Controller-Dual-Channel-Bridge-Driver-Module/dp/B07D2948RZ>
- [12] *Raspberry Pi 3 to DBH12V Motor Controller*, raspberrypiforum. [Online]. Available: <https://forums.raspberrypi.com/viewtopic.php?t=241060>
- [13] *Datasheet Raspberry Pi 3b+*, Raspberry Pi. [Online]. Available: <https://datasheets.raspberrypi.com/rpi3/raspberry-pi-3-b-plus-product-brief.pdf>
- [14] *ledstripaccubatterij12vmetoplader*, ledstripkoning. [Online]. Available: <https://www.ledstripkoning.nl/speciale-leds/led-strips-op-batterij/batterijen-accus-en-accessoires/led-strip-accubatterij-12v-met-oplader>
- [15] *datasheet powerbank rs pro*, rspro. [Online]. Available: <https://docs.rs-online.com/b019/0900766b81588796.pdf>
- [16] *Guidance UNOSQUARE PIGPIO*, PiGpio. [Online]. Available: <http://abyz.me.uk/rpi/pigpio/>
- [17] H.-Y. Chung, C.-C. Hou, and Y.-S. Chen, "Indoor intelligent mobile robot localization using fuzzy compensation and kalman filter to fuse the data of gyroscope and magnetometer," *IEEE Transactions on Industrial Electronics*, vol. 62, no. 10, p. 6436–6447, Oct. 2015. [Online]. Available: <http://dx.doi.org/10.1109/TIE.2015.2416692>
- [18] W. Lv, Y. Kang, and J. Qin, "Indoor localization for skid-steering mobile robot by fusing encoder, gyroscope, and magnetometer," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 49, no. 6, p. 1241–1253, Jun. 2019. [Online]. Available: <http://dx.doi.org/10.1109/TSMC.2017.2701353>
- [19] J.-x. Wang and X. Cui, "Rotary encoder based self-positioning method for mobile robot," in *2018 5th International Conference on Information Science and Control Engineering (ICISCE)*. IEEE, Jul. 2018. [Online]. Available: <http://dx.doi.org/10.1109/ICISCE.2018.00111>
- [20] T. A. Lima, M. Davi do Nascimento Forte, F. G. Nogueira, B. C. Torrico, and A. R. de Paula, "Trajectory tracking control of a mobile robot using lidar sensor for position and orientation estimation," in *2016 12th IEEE International Conference on Industry Applications (INDUSCON)*. IEEE, Nov. 2016. [Online]. Available: <http://dx.doi.org/10.1109/INDUSCON.2016.7874573>
- [21] D. Bartholomew, "Mariadb mysql postgresql and sqlite3 comparing command line interfaces," *Linux Journal*, vol. 2011, no. 212, p. 3, 2011. [Online]. Available: <https://dl.acm.org/doi/fullHtml/10.5555/2123870.2123873>
- [22] M. Biehl, *API architecture*. API-University Press, 2015, vol. 2. [Online]. Available: https://books.google.com/books?hl=nl&lr=&id=6D64DwAAQBAJ&oi=fnd&pg=PA15&dq=WHY+use+api&ots=zb3N9sNe6r&sig=boRyyCBXyg-y81_D3TbFJu2tzJQ
- [23] J. Price, *Mastering Database Programming*. John Wiley & Sons, 2006. [Online]. Available: https://books.google.com/books?hl=nl&lr=&id=tKytuTyw3boC&oi=fnd&pg=PR7&dq=reading+database+with+c%23&ots=yYTWlZSl66&sig=9WNFe6bT1rz8ChdLqTLE_pE56Lc
- [24] Microsoft, "Internet of things dotnet," <https://dotnet.microsoft.com/en-us/apps/iot>, [Accessed 2024-06-16].
- [25] A. Singh, P. Chawla, K. Singh, and A. K. Singh, "Formulating an mvc framework for web development in java," in *2018 2nd International Conference on Trends in Electronics and Informatics (ICOEI)*. IEEE, May 2018. [Online]. Available: <http://dx.doi.org/10.1109/ICOEI.2018.8553746>
- [26] P. Waher, *Mastering Internet of Things: Design and create your own IoT applications using Raspberry Pi 3*. Packt Publishing Ltd, 2018. [Online]. Available: <https://books.google.com/books?hl=nl&lr=&id=RLZTDwAAQBAJ&oi=fnd&pg=PP1&dq=iot+c%23+rasp&ots=q6zFa3x4ek&sig=BQ3PHwhVMeAH7671MDQlke31nWo>
- [27] Z. Lu and X. Liu, "Iot application development based on java and raspberry pi," in *2021 IEEE 12th Annual Information Technology, Electronics and Mobile Communication Conference (IEMCON)*. IEEE, Oct. 2021. [Online]. Available: <http://dx.doi.org/10.1109/IEMCON53756.2021.9623242>
- [28] J. Kanjilal, *ASP NET Web API*. Packt Publishing, 2013. [Online]. Available: <https://www.uploadbag.com/ofiles/bb04356fbcd1771e0e2c7a37264173d/ASP.NET-Web-API.pdf>
- [29] J. Trivedi, *Building a Web App with Blazor and ASP. Net Core: Create a Single Page App with Blazor Server and Entity Framework Core*. BPB Publications, 2020. [Online]. Available: <https://books.google.com/books?hl=nl&lr=&id=pCvxDwAAQBAJ&oi=fnd&pg=PP18&dq=canvas+in+blazor&ots=tuEXmjipVvk&sig=d1VKEhk4XIA7Nl3Sp6dm415iK4>
- [30] S. Furber and A. Wilson, "The acorn risc machine — an architectural view," *Electronics and Power*, vol. 33, no. 6, p. 402, 1987. [Online]. Available: <http://dx.doi.org/10.1049/ep.1987.0249>
- [31] R. Minhas and R. Bhatt, "Introduction to the dotnet," 2022. [Online]. Available: <http://www.ir.juit.ac.in:8080/jspui/bitstream/123456789/3699/1/Introduction%20to%20the%20Dotnet.pdf>
- [32] A. Andersson Tholin, "Evaluating remote and local web rendering of real-time interactive 2d graphics using blazor," 2020. [Online]. Available: <https://www.diva-portal.org/smash/get/diva2:1443507/FULLTEXT01.pdf>

- [33] Google, “Canvaskit skia webassembly,” <https://skia.org/docs/user/modules/canvaskit/>, [Accessed 2024-06-16].
- [34] T. Polák, “Využití webassembly pro vývoj webových aplikací.” [Online]. Available: <https://theses.cz/id/fipshh/STAG95944.pdf>
- [35] M. Nash, *Java Frameworks and Components: Accelerate Your Web Application Development*. Cambridge University Press, 2003. [Online]. Available: https://books.google.com/books?hl=nl&lr=&id=HgW-GqsMRiMC&oi=fnd&pg=PR11&dq=Java+Web+Application&ots=yfM6yfk_JR&sig=MeapdPA1tDw9hvp78j7PIFMR0cM
- [36] aspose, “Edit html5 canvas in java,” <https://docs.aspose.com/html/java/edit-html5-canvas/>, [Accessed 2024-06-16].
- [37] Google, “Skcanvas class reference,” <https://api.skia.org/classSkCanvas.html>, [Accessed 2024-06-16].

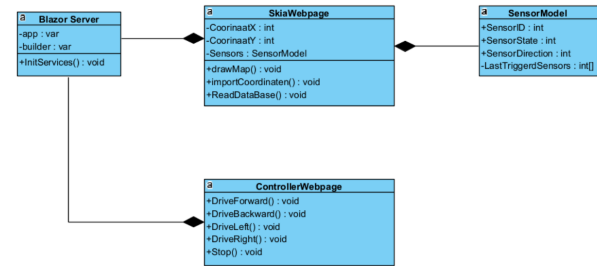


Figuur 3. Vernieuwde UML-klassediagram van de (object georiënteerde) webserver en webapplicatie.

BIJLAGEN



Figuur 1. Blokschema van systeemontwerp hardware.



Figuur 4. Aanvankelijke UML-klassediagram van de (object georiënteerde) webserver en webapplicatie.

Specificaties	Microswitch sensor
Spanningsbereik	Minimaal 3 Volt DC

Tabel II
TECHNISCHE SPECIFICATIES VAN DE MICROSWITCH SENSOR [10].

Specificaties	DBH-12V Motorcontroller
Interne weerstand	12 mΩ
Stroombereik	0 tot 30 Ampère
Spanningsbereik Motor	5 tot 12 Volt DC
Spanningsbereik Logica	3 tot 12 Volt DC
Impedantie belasting	200 kHz
PWM Duty Cycle	0% tot maximaal 98%

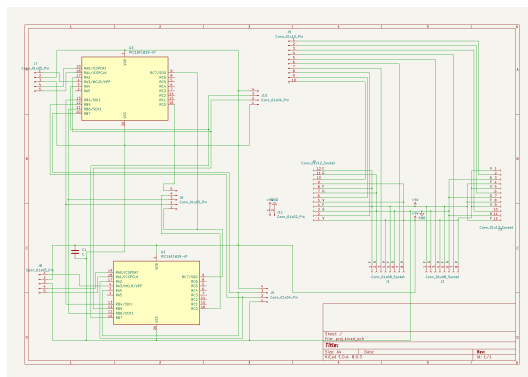
Tabel III
TECHNISCHE SPECIFICATIES VAN DE DBH-12V MOTORCONTROLLER [11].

Specificaties	Raspberry Pi 3b+
Processor	Broadcom BCM2837B0, Cortex-A53 1.4GHz
Geheugen	1GB
Connectiviteit	2.4 GHz and 5 GHz IEEE 802.11b/g/n/ac
Video en geluid	HDMI, MIPI DSI, MIPI CSI, stereo, video
Multimedia	H.264, MPEG-4, OpenGL
SD-kaart ondersteuning	Micro SD for operating system and data
Input Power	5V/2.5A DC micro USB, 5V DC GPIO, PoE
Werking temperatuur	0 – 50°

Tabel IV
TECHNISCHE SPECIFICATIES VAN DE RASPBERRY PI 3B+ [13].

Specificaties	Infrarode sensor
Effectieve meetbereik	2 tot 30 centimeter
Beschikt een potentiometer	Ja
Spanningsbereik	3 tot 5 Volt DC
Detectie hoek	35°

Tabel I
TECHNISCHE SPECIFICATIES VAN DE INFRARODE SENSOR [9].



Figuur 2. Schematische weergave (CAD-tekening) van het sensorhub waarbij twee microcontroller sockets beschikbaar zijn.

Specificaties	Accu Pack 12V
<i>Type batterij</i>	Lithium-ion
<i>Lengte batterij</i>	118 mm
<i>Breedte batterij</i>	71 mm
<i>Dikte batterij</i>	40 mm
<i>Gewicht batterij</i>	472 gram
<i>Capaciteit</i>	6600mAh
<i>Uitgangsvermogen (watt)</i>	82W
<i>Uitgangsspanning</i>	12V DC
<i>Aantal laadcycli</i>	500

Tabel V

TECHNISCHE SPECIFICATIES VAN DE 12 VOLT ACCU PACK [14].

Specificaties	5V powerbank
<i>Type batterij</i>	Lithium Polymer
<i>Type connector</i>	Micro USB
<i>Diepte batterij</i>	15.8 mm
<i>Breedte batterij</i>	66 mm
<i>Gewicht batterij</i>	472 gram
<i>Lengte batterij</i>	102 mm
<i>Max. uitgangsstroom</i>	2.1 A
<i>Capaciteit</i>	5000 mAh
<i>Gewicht</i>	154 g
<i>Uitgangsspanning</i>	5V DC
<i>Inputspanning</i>	5V DC

Tabel VI

TECHNISCHE SPECIFICATIES VAN DE 5 VOLT powerbank [15].