



KU Leuven

Departement Computerwetenschappen

P&O: COMPUTERWETENSCHAPPEN

Tussentijds verslag

Team:
Indigo

WANDER BAVIN
VINCE GOOSSENS
DIMITRI JONCKERS
SUNIL TANDAN
WOUT VEKEMANS

Academiejaar 2013 – 2014

Samenvatting

Dit rapport documenteert onze analyse en oplossing van het volgende probleem : de constructie en operatie van een zeppelin in wedstrijdverband. De zeppelin maakt gebruik van een gekend grondplan in roostervorm voor navigatie. De positie van de zeppelin wordt bepaald door een algoritme gebaseerd op beeldherkenning. Via het rooster moet hij sneller dan een andere zeppelin naar een bepaalde positie vliegen, en een andere, nog onbekende opdracht uitvoeren. Beide zeppelins dienen met een RabbitMQ-server te communiceren om informatie uit te wisselen met hun sturende laptop, en met elkaar. Een GUI dient de toestand van het speelveld met beide zeppelins te visualiseren. Al deze functionaliteiten worden geïmplementeerd in Java. Op dit moment staan alle delen van de software nog in de steigers, maar werkende functies zijn niet veraf.

Inhoudsopgave

1	Inleiding	2
2	Beschrijving materiaal en bouw zeppelin	2
3	Testen	4
3.1	Afstandssensor	4
3.2	Camera en image processing	4
3.3	Motoren	4
4	Algoritmes	4
4.1	Verticale bewegingen	4
4.2	Horizontale bewegingen	5
5	Software	5
6	Besluit	5
A	Beschrijving van het proces	6
B	Beschrijving van de werkverdeling	6
C	Kritische analyse	6

1 Inleiding

Dit tweede deel van de bachelorproef draait nog steeds rond het aansturen van een zeppelin op basis van een Raspberry Pi.

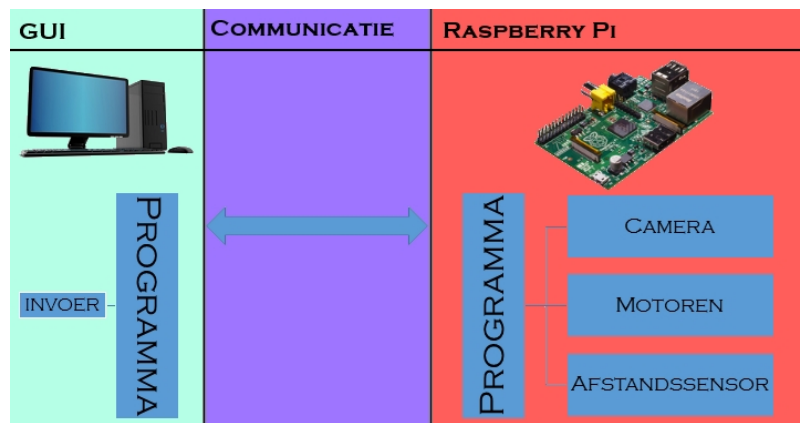
Fysisch ontwerp

De zeppelin bestaat uit een houten frame waaraan 2 heliumballonnen ($\varnothing 90$ cm) vastgemaakt zijn. Aan het frame zijn een camera en een afstandssensor vastgemaakt, die beiden naar beneden gericht zijn. Zowel de camera als de afstandssensor zijn verbonden met een Raspberry Pi die in het frame zit ingebed. Het geheel bevat drie propellers: twee voor horizontale bewegingen en één voor verticale bewegingen.

Software ontwerp

De software die inwendig op de Pi draait en op de client-pc voor image processing en weergave van de GUI is geschreven in Java. De communicatie tussen de zeppelin en de client gebeurt via een server. (Zie figuur 1).

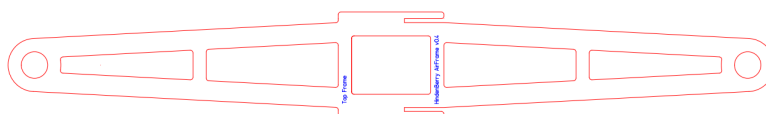
TODO !!!! schema figuur



Figuur 1: Architectuur

2 Beschrijving materiaal en bouw zeppelin

De zeppelin bestaat uit een frame waaraan alle onderdelen zijn vastgemaakt. Hierop worden onder andere de 3 propellers bevestigd. Twee hiervan dienen om naar links en rechts te draaien. Om vooruit te bewegen worden deze samen geactiveerd met dezelfde kracht. Deze propellers bevinden zich aan de uiteindes van de vleugels. De derde propeller, om de zeppelin te laten stijgen, is naar beneden gericht. De propellers kunnen op volle kracht of door middel van pwm¹ worden aangestuurd (in 2 richtingen). Met deze techniek is het mogelijk om naast de richting ook de kracht van de motor in te stellen. Het is slechts mogelijk om één propeller aan te sturen met pwm. We hebben hiervoor de motor voor verticale bewegingen genomen. Zo is het mogelijk de hoogte accuraat te regelen.



¹en.wikipedia.org/wiki/Pulse-width_modulation

de motor geen kracht naar boven zet. Als gewicht gebruiken we zout, zodat het mogelijk is om nauwkeurig het gewicht te regelen.

3 Testen

Om er zeker van te kunnen zijn dat het aansturen van de zeppelin correct gebeurt, is het nodig dat de componenten getest worden. In deze sectie volgt hierover meer informatie.

3.1 Afstandssensor

Voor gedetailleerde gegevens verwijzen we naar het verslag onze zeppelin van het eerste semester (versie 2). Hier hadden we gemerkt dat een enkele waarde van de sensor niet noodzakelijk de correcte afstand weergeeft. Daarom houdt de afstandssensor de laatste 10 gemeten waardes bij. De huidige hoogte wordt gegeven als het gemiddelde van deze waardes (rolling median techniek). Het interval tussen twee metingen hebben we kunnen terugbrengen tot 20 ms.

3.2 Camera en image processing

3.3 Motoren

In het eerste semester hebben we uitgebreide testen gedaan over de testen en de mogelijkheden van pwm. Het werd duidelijk dat het nodig ging zijn om constant de hoogte te controleren en bij te sturen. Daarnaast bleek ook dat horizontale bewegingen moeilijk exact kunnen worden uitgevoerd, omdat de zeppelin zeer gevoelig is voor allerlei veranderingen van de windomstandigheden in de omgeving: een andere zeppelin die beweegt, de airco,

4 Algoritmes

Voor het tweede deel was er een geheel nieuw bewegingsalgoritme nodig. Er moet gebruik gemaakt worden van de camera-input om de positie bij te sturen. Aan de hand van de guren op de vloer kan er genavigeerd worden. BLABLABLABLABLABLA HIER KOMT HET NIEUWE BEWEGINGSALGORITME

4.1 Verticale bewegingen

Om naar een bepaalde hoogte te stijgen, maken we gebruik van een PID-algoritme². Hierbij gaan we op basis van de huidige fout in hoogte, bepalen of de zeppelin moet stijgen of dalen en met welke kracht. Daarnaast wordt rekening gehouden met de afgeleide, om toekomstige veranderingen te voorspellen. Tenslotte is er de integraal, die fouten uit het verleden voorstelt. Op basis hiervan wordt een pwm-waarde voor de motor gegeven. Het algoritme wordt aangepast aan het gewicht van onze zeppelin en de kracht van de gegeven motoren.

De output wordt berekend op basis van deze formule:

$$\text{output} = K_p \cdot \text{error} + K_i \cdot \text{integral} + K_d \cdot \text{derivative}$$

Hierin zijn K_p , K_i en K_d constanten die we hebben moeten bepalen. Eerst hadden we enkel rekening gehouden met de huidige error ($K_i = K_d = 0$), maar dit bleek er voor te zorgen dat de zeppelin veel te snel naar een bepaalde hoogte gaat en er dan ver boven of onder gaat. We hebben dit opgelost door K_d te verhogen. Door deze groot te maken, gaat de zeppelin veel rustiger naar de opgegeven hoogte en gaat hij er niet voorbij. Bij de start van het tweede semester hebben we de waardes van deze constanten opnieuw getuned, aangezien er nieuwe volle ballonnen gegeven waren.

Er is een HeightController die dit PID-algoritme implementeert, en die er voor gaat zorgen dat de zeppelin zijn hoogte behoudt of naar een gevraagde hoogte gaat. Deze gaat om de 0.1 s de hoogte

²http://www.csimn.com/CSI_pages/PIDforDummies.html

controleren en de pwm-waarde bijsturen.

4.2 Horizontale bewegingen

TODO!!!!!!

5 Software

TODO!! afbeelding

De User Interface stelt de gebruiker in staat om vanaf een client-pc verbinding te maken met de server die het speelveld en de zeppelins controleert. De GUI geeft dan informatie weer over het speelveld en de locatie van de zeppelins hierop, en de huidige hoogtes van beide zeppelins.

De eerste tab ('overview') toont de huidige hoogtes en de toestand van de propellers van de eigen zeppelin. Op de kaart van het speelveld worden de locaties van de zeppelins weergegeven. Daarnaast wordt de opdracht van de zeppelin getoond en is er een overzicht van de laatste berichten die zijn uitgewisseld tussen de client en de zeppelin.

De tweede tab geeft een uitgebreider overzicht van alle informatie die wordt uitgewisseld tussen de GUI en zeppelin, met een indicatie van de tijd. Er is een mogelijkheid om berichten te filteren op type.

Voor het tweede deel van de opdracht was er een vorm van image recognition. Hiervoor hebben we gebruik gemaakt van de Java-library OpenCV. Hiermee kan detectie van vormen en kleuren worden geïmplementeerd. Connectie Ook de verbinding is helemaal veranderd in het tweede deel van de opdracht. De twee zeppelins moeten nu communiceren via een server. Die maakt gebruik van de RabbitMQ library.

6 Besluit

A Beschrijving van het proces

Dit onderdeel maakt geen deel uit van dit verslag.

B Beschrijving van de werkverdeling

Hieronder is een tabel te vinden met de gewerkte uren binnen en buiten de sessies:

Overzicht:	Dimitri Jonckers	Wander Bavin	Wout Vekemans	Sunil Tandan	Vince Goossens
09/02 - 15/02					
Totaal					

C Kritische analyse

Dit onderdeel maakt geen deel uit van dit verslag.