

Autonomous F1/10-car using Artificial Intelligence

Jens de Hoog*, Thomas Huybrechts*[§], Peter Hellinckx*[§]

jens.dehoog@student.uantwerpen.be, thomas.huybrechts@uantwerpen.be, peter.hellinckx@uantwerpen.be

*Department of Applied Engineering, Electronics-ICT, University of Antwerp, Belgium

[§]IDLab, Department of Applied Engineering, University of Antwerp - iMinds, Belgium

Abstract—Insert abstract of the paper.

I. INTRODUCTION

II. F1/10-CAR

A. Global

Ik vertel over de wagen in het algemeen en welke sensoren eropzitten. Dit gebeurt heel kort. Ook vertel ik over de globale structuur van ROS en welke belangrijke nodes er gebruikt worden.

B. Prerequisites

In dit deel vertel ik over het traag rijden en het rijden tegen constante snelheid.

- Slow accurate driving: er wordt een beetje aandacht besteedt aan de twee approaches en waarom ze wel/niet werkten
- Ik vertel kort waarom het rijden op softwareniveau (met IMU) niet werkten en hoe het met de RPM-sensor is opgelost.

III. ALGORITHMS

A. Global

Hierin vertel ik over de verschillende approaches en wat ze verschillend maakt.

B. Fixed path

Literature

In dit deel leg ik verschillende globale algoritmes naast elkaar. Ook vertel ik dat de Navigation Stack aanwezig is in ROS en dat deze daa

Navigation Stack

Hierin vertel ik dat er een implementatie bestaat in het ROS-framework, hoe deze is opgebouwd (subnodes, lokale / globale planner, AMCL, ...), wat er is foutgelopen bij het opzetten van de Stack en wat de oplossingen daarvoor waren. Ook vertelt dit deel over het feit dat de Stack in principe gemaakt is voor een differentiële of holonomische sturing. Oplossingen worden daarvoor aangereikt, samen met de uitleg waarom die oplossingen goed/slecht waren. Het foutgelopen+oplossingen-deel is beknopt. De oplossingen worden aangereikt in een korte maar duidelijke manier.

Resultaten

Dit deel vertelt over het verloop van het navigeren

d.m.v. de Stack. Er komen fotos aan bod van het uitgestippelde pad, de auto die tegen de muur zit, ...

C. Waypoints

Literature

In dit deel komen verschillende algoritmes naar boven die in de literatuur te vinden zijn. Ook wordt er gezegd waarom deze algoritmes niet goed genoeg zijn voor mijn doel en dat ik daarom zelf een algoritme heb ontwikkeld.

Concept

Deze paragraaf legt het algoritme uit, samen met alle edge cases (wat als een muur wegvalt, wat als ik een kruispunt tegenkom, ...).

Resultaten

Dit deel beschrijft de resultaten. Hierin komt ook aan bod dat het algoritme in het algemeen goed werkt, maar dat de LiDAR-sensor gevoelig is aan de soort reflectie (zwarte voorwerpen hebben minder reflectie) en dat er daardoor veel afstelwerk vereist was. Dit komt ook aan bod in further research.

D. Optimisation Algorithm

Literature

Ik leg verschillende algoritmes naast elkaar en maak een kleine vergelijkende studie. Ik leg ook uit waarom ik een bepaald algoritme heb gekozen.

Concept

Het gekozen algoritme wordt in meer in detail besproken.

Resultaten

De resultaten van dit algoritme wordt besproken.

IV. COMPARATIVE STUDY

De hierboven voorgelegde approaches worden met elkaar vergeleken en er worden verschillende toepassingen opgesomd waarin elk algoritme faalt of uitblinkt.

V. CONCLUSION

De conclusie wordt gemaakt.

VI. FURTHER RESEARCH

In dit deel wordt kort besproken wat er allemaal nog gedaan kan worden van research. Er kan bijvoorbeeld gekeken naar worden optimalisaties voor de Navigation Stack en meer specifiek voor Ackermannsteering. Ook kan er gekeken worden naar betere herkenning van hoeken en inhammen voor het tweede algoritme. Als laatste zouden er betere optimalisatie-algoritmes onderzocht kunnen worden, en met of zonder neural network.