

Robots Kinematica 2021 S2

Joost Kraaijeveld, Chris van Uffelen en Richard Holleman

Monday 21 March, 2022

1 Inleiding

De tweede beroepsopdracht van WoR-R bestaat uit twee gedeelten.

1. Het individueel uitprogrammeren van onderstaande opdracht ¹. Het technisch en functioneel werkend krijgen van deze opdracht is een voorwaarde om deel te mogen nemen van het tweede gedeelte. Technisch en functioneel moet de oplossing van aanvaardbaar niveau zijn. Dit betekent dat alle requirements geïmplementeerd moeten zijn maar dat het niet onoverkomelijk is als er incidenteel nog een bugje in zit.
2. Het schrijven van een peer-reviewrapport van de uitwerking van deze opdracht van een andere student.

Dit rapport wordt beoordeeld als voldaan of niet-voldaan.

2 Opdracht

1. Ga uit van de basisversie van Robotwereld. In de huidige master zijn heel veel warnings opgelost in RobotWorld zelf. Als je bovendien gebruik maakt van het bijgeleverde autotools build systeem dan heb je vrijwel geen last van allerlei Boost en wxWidgets warnings bij het compileren of het gebruik van CPPCheck.
2. Vergroot de wereld tot 1024x1024 pixels.
3. Voeg ten minste 4 muren van tenminste 100 pixels toe aan de wereld.
4. De robot heeft kennis van de kaart van de wereld.

¹Het is toegestaan om met elkaar te overleggen of raad te vragen aan docenten, maar het is *niet* toegestaan om code van elkaar over te nemen, ook niet na het slim hernoemen van classes, functies of variabelen, dan wel wat wijzigingen in de volgorde van functies.

5. De robot heeft een positie en oriëntatie die gebruikt wordt voor het bepalen van de route, het rijden en als input voor de sensoren (van waaruit er gemeten moet worden). Daarnaast moet de robot een 'believe' positie en oriëntatie krijgen.
6. Het bepalen van de route met A-star en het automatisch rijden naar het doel kan beschouwd worden als het handmatig besturen / laten rijden van de robot.
7. De robot maakt gebruik van de Kalmanfilter op basis van kompas en odometer om een overtuiging van zijn positie en oriëntatie te vormen.
8. De robot maakt gebruik van het Particlefilter op basis van kompas en lidar om een overtuiging van zijn positie en oriëntatie te vormen.
9. De robot ontwikkelt een overtuiging van de werkelijkheid volgens twee methodes (zie voorgaande twee items). Voeg aan de GUI van robotwereld een optie toe waarmee de gebruiker kan selecteren of de robot gebruik maakt van de overtuiging verkregen met Kalmanfilter of die uit het Particlefilter om een keuze te maken voor de control update.
10. De controlupdate gaat gepaard met onzekerheid. Bij een snelheid van 10 pixels per stap is de kans 0.7 dat deze uitkomt in een van de pixels in het 3x3 vierkant rondom de doelpixel. De kans is 0.3 dat deze uitkomt in één van de overige pixels in het 5x5 vierkant rondom de doelpixel.
11. Het kompas van de robot bepaalt de rijrichting van de robot met een configureerbare standaarddeviatie. De uitgangswaarde van de standaarddeviatie is 2 graden.
12. De odometer van de robot meet de afgelegde weg van de robot met een configureerbare standaarddeviatie. De uitgangswaarde van deze standaarddeviatie is 1 pixel per stap van 10 pixels.
13. De lidar van de robot meet per ronde 180 keer (dat is iedere 2 graden). De afstandsmeting van iedere "beam" heeft een configureerbare standaarddeviatie. De uitgangswaarde van deze standaarddeviatie is 10 pixels.
14. Laat de robot naar een willekeurig doel in de wereld rijden. Plot in de robotwereld de werkelijk afgelegde weg, de overtuiging van de afgelegde weg volgens het Kalmanfilter, en de overtuiging van de afgelegde weg volgens het Particlefilter.

2.1 Compilatie en run-time vereisten

Voor de beoordeling van de toelaatbaarheid gelden de volgende z.g. "knock-out"-criteria. Als je niet aan deze onderstaande criteria voldoet dan is de inzending sowieso niet toelaatbaar:

- Het programma moet als een standaard C++-programma in Eclipse compileren op een “stock” Linux Debian Testing (Bullseye) of Ubuntu 20.04 LTS (Focal Fossa) distributie met tenminste GCC 9.3.
- Als je er voor kiest om een op een andere manier dan met de bijgeleverde (eventueel aangepaste) Autotools te compileren, e.g. cmake of een andere vorm van compilatie-script, dan moet je bron-code een duidelijke compilatie-instructie of script bevatten, in een README-bestand of anderszins gedocumenteerd. Denk eraan dat ik dom, oud en opvallend vaak chagrijnig ben. Ik ben weinig tot niet geneigd na te denken over, of me bezig te houden met, het debuggen van andermans compilatie-instructies...
- De repo mag geen overbodige bestanden bevatten. Overbodige bestanden zijn bestanden die door tools gegenereerd worden (o.a. maar niet beperkt tot, object-bestanden en executables, of bestanden die typisch lokaal zijn, o.a. maar niet beperkt tot, Eclipse-project bestanden of CMake-artefacten).

3 Inleveren

Het inleveren gaat als volgt. Allereerts: je mailt een link naar jouw repository, samen met een zip met daarin je repository, naar de docenten. De repository heeft minimaal 2 commits. 1) de initiele robotworld zonder aanpassing en 2) al jouw wijzigingen (idealiter commit je na elke significante wijziging). Ten tweede: de Git-repository met de code van het programma moet uiterlijk op het in ISAS aangegeven moment ingeleverd zijn op ISAS.

LET OP! Het werk moet volgens ISAS uiterlijk op donderdag 7 april ingelevert worden. Dit is al week 9. Daarom zal de review in week 1 gehouden worden.

Zodra de inzendingen op toelaatbaarheid zijn beoordeeld wordt meegedeeld wie aan de peerreview mogen meedoen. Dit gebeurt uiterlijk op de dinsdag van week 1.

4 Peerreview

De peerreview wordt gedaan op de vrijdag van week 1 van 12:30 tot 14:30. Nadere details met betrekking tot de peerreview volgen.