

Beroepsproduct Simulatie: Niet knoeien

1 Inleiding

In de course heb je geleerd om software te maken waarmee je eenvoudige actuatoren zoals robotarmen kunt ontsluiten en aansturen. Daar wil je natuurlijk goed gebruik van maken door een robotsysteem te bouwen die jouw je favoriete drankje kan aanreiken. Tijdens de ontwikkeling van het systeem en het testen van je planningsalgoritmen wil je natuurlijk niet je hardware beschadigen door continue bekertjes om te stoten met een nog niet geheel werkende robotarm. We hebben daarom behoefte aan een “virtueel bekertje water” waarmee veilig getest en geëxperimenteerd kan worden.

In dit document vind je de opdrachtomschrijving, de gestelde eisen met bijbehorende normering, en enige informatie met betrekking tot het inleveren van de opdracht.

Met betrekking tot het inleveren gelden de volgende eisen:

- Inleveren moet via iSAS vóór de via het toetsrooster bekend gemaakte datum en tijd.
- Alle onderdelen dienen gebundeld in één bestand te worden aangeleverd (zip, rar of tar-gzip)
- Documenten mogen in Word of pdf formaat worden aangeleverd.
- Losse plaatjes zijn alleen toegestaan indien er naar wordt verwezen in de documenten (alleen in PNG formaat).
- Werk in tweetallen

NB: Mogelijk kun je uitgenodigd worden je uitwerking mondeling toe te lichten.

2 Opdrachtbeschrijving

In deze opdracht moet je een Robot OS (ROS) package ontwikkelen dat virtuele bekertjes kan verzorgen. Andere ROS packages (bijvoorbeeld packages die je in het project ontwikkeld) kunnen er dan gebruik van maken. Ter ondersteuning van experimenten moet het bekertje informatie over zichzelf publiceren zoals snelheid, versnelling en kanteling.

Naast het bekertje ontwikkel je een demonstratie van een voorbeeldinteractie tussen een virtuele robotarm en een virtueel bekertje. Je zult hiervoor ook een virtuele implementatie van de AL5D controller moeten maken. Als laatste schrijf je een handleiding waarin je uitlegt hoe je deze demo kunt builden en uitvoeren en welke eisen je ermee demonstreert.

De volgende paragrafen beschrijven de aan uitwerking gestelde eisen en de daar aan gekoppelde puntentelling.

Eisen

De uitwerking van je beroepsproduct zal op de volgende punten beoordeeld worden:

Package (9 punten)

ID	Omschrijving	Prio	Punten
PA01	Alle code is gepackaged volgens de ROS directorystructuur	Should	3
PA02	Package is te bouwen met catkin/cmake op ROS Kinetic	Should	3
PA03	Alle code voldoet aan de ROS C++ Style Guide	Should	3

Virtuele servocontroller (15 punten)

ID	Omschrijving	Prio	Punten
VS01	De virtuele controller luistert naar een topic waarop string messages in het formaat van de SSC-32U controller	Must	3
VS02	De virtuele controller hierop door bijbehorende joint_state messages te publiceren	Must	5
VS03	De virtuele robotarm wordt gevisualiseerd in Rviz (een urdf model van de arm is beschikbaar)	Could	3
VS04	De virtuele robotarm gedraagt zich realistisch mbt tijdgedrag (servo's roteren kost tijd en gaat gelijdelijk)	Should	2
VS05	De virtuele robotarm kan op een willekeurige plaats in de virtuele world geplaatst worden	Should	2

Virtueel bekertje (36 punten)

ID	Omschrijving	Prio	Punten
VC01	Er kan op een willekeurige plek in de virtuele wereld een beker geplaatst worden	Should	4
VC02	Er kunnen meerdere bekertjes tegelijk geplaatst worden	Could	1
VC03	Publiceert een 3D visualisatie van het (de) bekertje(s) voor Rviz	Could	4
VC04	Detecteert de relevante punten van de gripper	Should	5
VC05	Visualiseert de gedetecteerde punten van de gripper	Could	2
VC06	Bepaalt wanneer de gripper het bekertje vastheeft	Must	5
VC07	Visualiseert wanneer de gripper de beker vastheeft	Could	2
VC08	Beker beweegt mee met de gripper (als hij vastgehouden wordt)	Must	4
VC09	Beker bepaalt en publiceert zijn snelheid	Should	2
VC10	Beker bepaalt en publiceert zijn versnelling	Could	1
VC11	Beker bepaalt en publiceert zijn hellingshoek(en)	Could	1
VC12	Snelheid en/of versnelling worden getoond met rqt_plot	Could	1
VC13	Beker is onderhevig aan zwaartekracht wanneer losgelaten	Could	2
VC14	Beker kan verschoven worden door met de gripper tegen de beker te duwen	Could	2

Demonstratie-infrastructuur (8 punten)

ID	Omschrijving	Prio	Punten
DI01	Demo wordt kan automatisch gerund worden met een roslaunch configuratie	Should	2
DI02	Locatie van de beker(s) wordt in de roslaunch configuratie bepaald	Could	2
DI03	Locatie van de arm van de arm wordt in de roslaunch configuratie bepaald	Could	2
DI04	Een demoscript stuurt over de tijd een sequentie van commando's naar de armcontroller	Should	2

Demonstratiehandleiding (14 punten)

ID	Omschrijving	Prio	Punten
DM01	Voldoet aan ICA controlekaart	Must	2
DM02	Beschrijft hoe (en onder welke voorwaarden) de code gebouwd kan worden	Must	4
DM03	Beschrijft stap voor stap hoe een (of meerdere) demonstratie(s) van de simulatie uitgevoerd kan worden	Must	4
DM04	Beschrijft per onderdeel welke criteria er mee aangetoond worden	Must	4

Ontwerpdocumentatie (18 punten)

ID	Omschrijving	Prio	Punten
DD01	Voldoet aan ICA controlekaart	Must	2
DD02	Beschrijft de structuur van de package	Should	4
DD03	Beschrijft de structuur en samenhang van de broncode	Should	4
DD04	Beschrijft hoe het gedrag van alle belangrijke componenten gerealiseerd is	Should	5
DD05	Beschrijft de API van alle publieke interfaces	Should	3

3 Beoordeling

Met de in het voorgaande beschreven eisen zijn in totaal 100 punten te verdienen. Als je niet aan **alle** must eisen voldoet krijg je maximaal 50 punten. Het cijfer is het aantal punten gedeeld door 10.

Als je denkt dat je een betere bekersistimulatie kunt maken, maar die niet precies de gestelde eisen afdekt, mag je een voorstel voor vervangende eisen doen. Wanneer je voorstel wordt goedgekeurd kun je op basis hiervan ook de punten krijgen.