

# Beroepsproduct Simulatie: Niet knoeien(herkansing)

Bram Knippenberg, Bas Lijnse, Jorg Visch  
6 oktober 2020

## 1 LET OP (HERKANSERS)

De hieronder beschreven opdracht is **alleen** bestemd voor studenten die deze opdracht moeten herkan- sen en een uitvoering van WoR van vóór 2019-2020 hebben gevolgd.

Volg je WoR in september 2019 (of later) voor het eerst, dan moet je de andere 'Simulatie-opdracht' doen. Zie [OnderwijsOnline](#).

Als je twijfelt, neem contact op met de uitvoerende docent van WoR World.

## 2 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 jou 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).
- De source code moet in de vorm van een git-repository (clone) worden aangeleverd.
- Documenten mogen alleen in PDF-formaat worden aangeleverd.
- Losse plaatjes zijn alleen toegestaan indien er naar wordt verwezen in de documenten (alleen in PNG-formaat).

*Iedereen moet een demonstratie geven van de gerealiseerde software, waarbij de uitwerking mondeling wordt toegelicht.*

## 3 OPDRACHTBESCHRIJVING

In deze opdracht moet je een Robot OS (ROS) package ontwikkelen dat virtuele bekertjes kan simuleren. Andere ROS packages (bijvoorbeeld packages die je in het project ontwikkelt) 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 bouwen en uitvoeren, en welke eisen je ermee demonstreert.

Hieronder worden de eisen beschreven 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 op ROS Melodic Morenia.	Must	2
PA03	De applicatie wordt gebouwd met C++ volgens de Object Oriented principes die je geleerd hebt bij eerdere courses.	Must	1
PA04	Alle code voldoet aan de ROS C++ Style Guide.	Should	3

### **VIRTUELE SERVO CONTROLLER (15 PUNTEN)**

ID	Omschrijving	Prio	Punten
VS01	De virtuele controller luistert naar een topic waarop string messages in het formaat van de <b>SSC-32U</b> worden geplaatst.	Must	3
VS02	De virtuele controller reageert op het topic (zie eis VS01) 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 op OnderwijsOnline).	Must	3
VS04	De virtuele robotarm gedraagt zich realistisch m.b.t. tijdgedrag (servo's roteren kost tijd en gaat geleidelijk).	Should	2
VS05	De virtuele robotarm kan op een willekeurige plaats in de virtuele wereld geplaatst worden.	Should	2

### **VIRTUEEL BEKERTJE (36 PUNTEN)**

ID	Omschrijving	Prio	Punten
VC01	Er kan op een willekeurige plek in de virtuele wereld een bekertje geplaatst worden.	Should	4
VC02	Er kunnen meerdere bekertjes tegelijk geplaatst worden.	Could	1
VC03	Publiceert een 3D-visualisatie van het bekertje voor Rviz.	Must	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 het bekertje vastheeft.	Could	2
VC08	Het bekertje beweegt mee met de gripper (als hij vastgehouden wordt).	Must	4
VC09	Bekertje bepaalt en publiceert zijn snelheid.	Should	2
VC10	Bekertje bepaalt en publiceert zijn versnelling.	Could	1
VC11	Bekertje bepaalt en publiceert zijn hellingshoek(en).	Could	1
VC12	Snelheid en/of versnelling worden getoond met <code>rgt_plot</code> .	Could	1
VC13	Bekertje is onderhevig aan zwaartekracht wanneer losgelaten.	Could	2
VC14	Bekertje kan verschoven worden door met de gripper tegen het bekertje te duwen.	Could	2

### **DEMONSTRATIE-INFRASTRUCTUUR (8 PUNTEN)**

ID	Omschrijving	Prio	Punten
DI01	Demo wordt automatisch gerund met een roslaunch-configuratie.	Should	2
DI02	Locatie van het bekertje wordt in de roslaunch-configuratie bepaald.	Could	2
DI03	Locatie van de arm in de wereld wordt in de roslaunch-configuratie bepaald.	Could	2
DI04	Een demoscrypt 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 één of meerdere demonstraties 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 (Nodes, topics, messages, et cetera).	Must	4
DD03	Beschrijft de structuur en samenhang van de broncode (class-diagrams, beschrijving, et cetera).	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

## 4 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 bekersistulatie kunt maken, maar die niet precies de gestelde eisen afdekt, mag je een voorstel voor vervangende eisen doen. Als je voorstel wordt goedgekeurd kun je op basis hiervan ook de punten krijgen.