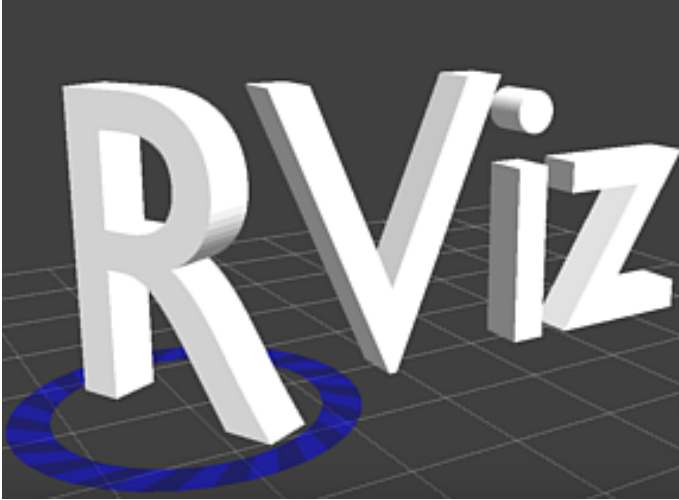


Robot Simulation



Thomas Maters
551773
WoR-World

Package structure

De package structuur ziet er als volgt uit:

- config (Hier zijn config file voor rviz te vinden)
- launch (Folder voor de launch file(s) om de applicatie makkelijk te starten)
- msg (Ros messages)
- src (Source code van het project)
- urdf (De urdf modellen voor rviz)

Build requirements

Om dit project te bouwen zijn de volgende vereisten nodig:

- ROS Lunar(minimaal)
- Boost v1.58
- OpenCV 2
- Gnu 6.2
- thread
- chrono
- regex

Code structure

De code is opgebouwd uit de volgende onderdelen:

- ssc32u_controller
- virtual_cup_node
- demo_node

De verbindingen van deze onderdelen wordt weergegeven in het bijgeleverde plaatje met de naam 'rosgraph.png'.

ssc32u_controller:

De 'ssc32u_controller' is verantwoordelijk om commando's in de vorm van het SSC32U protocol die binnen komen op het topic '/ssc32u_topic' om te zetten naar JointStates en die te publiceren op het '/joint_states' topic. De ssc32u_controller is geïmplementeerd door middel van regex om de commando's te parsen en daarna realistisch (linear) te interpoleren na de posities in het commando.

virtual_cup_node:

De 'virtual_cup_node' is verantwoordelijk voor het simuleren van een beker. Dit houdt in dat er collision detectie, realistische zwaarte kracht en physics wordt gesimuleerd. Dit wordt gedaan door middel van 'tf' en het berekenen van de [afstand van een punt tot een lijn](#).

De beker wordt gevisualiseerd door een marker message op het topic '/cup/cup_node_????' te sturen. Elke instantie van de node heeft een unieke naam zodat er meerdere bekertjes tegelijk gesimuleerd kunnen worden zonder problemen.

Ook worden gegevens van de beker gepubliceerd op het topic '/cup/cup_node_????_data'. Deze gegevens zijn de state van de cup, snelheid, versnelling, roll, pitch en yaw.

demo_node:

De 'demo_node' is verantwoordelijk om een kleine demo te geven van wat de simulatie kan. Dit doet hij door voorgeprogrammeerde SSC32U commando's op het '/ssc32u_topic' te sturen waar de ssc32u_controller dan weer naar luistert.

API

Er worden de volgende publieke interfaces aangeboden:

ssc32u_controller(ssc32u_parser_interface.h):

void ssc32uCommandReceived(const robotsimulation::ssc32u_command msg);

void sendJointStateMessage(const sensor_msgs::JointState& msg) const;

virtual_cup_node(virtual_cup_interface.h):

void sendCupData(const robotsimulation::cup_data& msg) const;

void sendMarkerData(const visualization_msgs::Marker& msg) const;

Build instructions

Om het project te builden moeten de volgende commando's uitgevoerd worden:

```
cd ../  
catkin_make
```

Run instructions

Om de demo te starten voer de volgende commando's uit in uw terminal:

```
source ./devel/setup.bash  
roslaunch robotsimulation default.launch demo:=true
```

Vervolgens dient in RVIZ de markers onder de '/cup' namespace toegevoegd te worden.

Om handmatige de simulatie aan te sturen dient uw de volgende commando uit te voeren:

```
source ./devel/setup.bash  
roslaunch robotsimulation default.launch  
rostopic pub /ssc32u_topic robotsimulation/ssc32u_command "command: '#0P1500T1500'"
```

Vervolgens dient in RVIZ de markers onder de '/cup' namespace toegevoegd te worden.
'#0P1500T1500' dient uw te vervangen met een geldig ssc32u commando.

Wilt uw de arm besturen met een GUI type de volgende commando's in uw terminal

```
source ./devel/setup.bash  
roslaunch robotsimulation default.launch gui:=true
```

Vervolgens dient in RVIZ de markers onder de '/cup' namespace toegevoegd te worden.

Wilt u meerdere bekens tegelijk simuleren, voer de volgende commando's uit op een nieuwe terminal:

```
source ./devel/setup.bash  
roslaunch robotsimulation virtual_cup_node [x] [y] [z]
```

De parameters 'x', 'y' en 'z' zijn optioneel.

Vergeet niet de nieuwe cup marker in RVIZ onder de '/cup' namespace toe te voegen.

Wilt u meerdere bekertjes bij het starten van de launch file?

Open dan het bestand 'default.launch' te vinden in 'roboticsimulation/launch' met een tekst editor.
Plaats dan de volgende regel zo vaak in de launch file als u bekertjes wilt simuleren

```
<node name="$(anon *****)" pkg="roboticsimulation" type="virtual_cup_node" launch-  
prefix="xterm -e" args="0.32 0.0 0.5"/>
```

Vervang bij elke nieuwe beker '*****' in een unieke naam zoals 'cup_node1' of 'cup_node2'.