Tematica proiect SCR

# Instalare tool-uri de dezvoltare a proiectului.

Pentru a executa mediul de simulare in scopul alcaturirii proiectului se vor urmari uramatorii pasi:

1. Se instaleaza aplicatia Docker Desktop de la adresa <https://docs.docker.com/desktop/setup/install/windows-install/>
2. Aceasta trebuie executata si mentinuta in acest stagiu pe tot parcursul experimentelor de simulare
3. Se descarca arhiva “scr\_project.zip” si se dezarhiveaza

In interiorul acesteia se vor afla 1 director si 3 fisiere:

src - directorul unde se afla configurarea pachetelor ROS pentru a interactiona cu mediul de simulare;

Dockerfile - fisierul de configurare a mediului izolat container docker pentru a executa programul generat si pentru a vizualiza datele vehiculate in ecosistemul ROS

run\_scr.bat - fisier executabil pentru a realiza compilarea si intrarea in sistemul de operare al masinii virtuale docker

run\_sim.bat - fisier executabil pentru rularea mediului de simulare

1. Punerea in executie a mediului de simulare

Se va rula initial executabilul “run\_sim.bat” ce va deschide o linie de comanda ce va fi necesar de a fi mentinuta deschisa pe tot parcursul simularii.

Vizualizarea interfetei grafice a mediului de simulare se face prin accesarea adresei <http://localhost:8080/vnc.html> unde se va apasa butonul central “Connect”.

In interfata grafica se afla, in zona centrala, fereastra de randare unde este afisat mediul de lucru impreuna cu robotul si obstacolele dinamice. Pentru a muta zona de urmarire se pot folosi urmatoarele actiuni:

* <Ctrl + LeftClick> miscare orbitala a camerei in spatiul simularii
* <Ctrl + MiddleClick> translatare a camerei in spatiul simularii
* <ScrollWheel> apropiere/departare a camerei in spatiul simularii

De asemenea in zona dreapa se afla la dispozitie butonul “Reset Simmulation”, de unde se poate realiza repozitionarea robotului in postura initiala fara a fi necesara resetarea mediului de simulare.

Note: Posibile probleme si rezolvari:

* Pentru a termina procesul de simulare in mod corect, se face prin inserarea a comenzii de tastatura <Ctrl + c>, in caz contrar va trebui inchis din aplicatia Docker Desktop ->Containers/Apps, iar container-ul “running” trebuie oprit prin al 3-lea buton numit stop.

1. Interactiunea cu mediul de simulare

Interactiunea cu mediul de simulare se face prin executarea “run\_scr.bat” ce realizeaza procedeul de compilare si deschide mediul masinii virtuale docker ce are preinstalate toate facilitatile de depanare ROS si se pot observa structurile topic-urilor expuse de catre simulare.

Pentru a rula noduri ROS scrise in limbajul python care sa preia mesaje de sensoristica si sa trimita comenzi de miscare robotului, se realizeaza prin modificarea fisierului “scr.py” ce se afla la calea relativa ”src\scr\_pkg\scr\_pkg\scr.py” din arhiva atasata. Modificarea codului poate fi facuta prin orice mediu IDE preferat, insa cea mai mare interconectare o are Visual Studio Code.

Dupa modificarea codului nodul poate fi lansat din linia de comanda al masinii virtuale lansatae de catre scripul “run\_scr.bat” folosind in ordine urmatoarele comenzi:

* source install/setup.bash
* ros2 run scr\_pkg scr\_node

# Task-uri de indeplinit in cadrul proiectului:

Proiectul presupune dezvoltarea ***modelului*** controlului cinematic pentru un robot diferential ce are ca task parcurgerea unui hol pana la iesirea din acesta evitand ciocnirile cu mediul si obstacole.

Robotul porneste din postura x = 0, y = 0, theta = pi/6 si trebuie sa ajunga in siguranta la starea dorita x = 10, y=0, theta = -pi/6. Initial acesta va trebui sa strabata un hol ce nu are obstacole urmand sa ocoleasca obstacolele statice generate aleator pentru a ajunge la destinatie pe baza informatiilor de pozitie si a sistemelor LiDAR.

Implemetarea se va face in limbajul python folosid mediul pus la dispozitie pe edu.tuiasi.ro sau pe email.

Simularea expune in mediul ROS 3 topic-uri principale:

* /cmd\_vel
* /laser
* /pose

Folosind topicul “/cmd\_vel” se realiza miscarea robotului pe baza comenzilor carteziene de viteza. Tipul de mesaj folosit este “geometry\_msgs/Twist”, unde pentru robotul nostru diferential, vor fi folosite doar componentele “linear.x” si “angular.z” cu scopul de a realiza translatie pe axa X, respectiv giratie pe axa Z fata de coordonatele locale ale robotului. Controller-ul intern functioneaza la frecventa de 500 Hz.

Din topicul “/laser” se obtin datele inregistrate de catre sistemul simulat LiDAR, ce este folosit pentru a evita coliziunile cu mediul. Acest topic foloseste tipul de mesaj “sensor\_msgs/LaserScan” ce descrie atat dispunerea razelor cat si limitele minime si maxime de detectie ale obstacolelor. In simulare razele de detctie sunt in numar de 16 dispuse 360 de grade in jurul robotului, cu o distanta de detectie minima de 0.25 metri si maxima de 2 metri. Frecventa datelor furnizate pe topic este de 10 Hz.

Topicul “/pose” descrie postura robotului fata de sitemul de coordonate global al simularii. Descrierea postrurii se face prin mesjul “geometry\_msgs/Pose”, unde pozitia este descrisa cartezian in structura “position” si orientarea sub forma de quaternion in structura “orientation”. Publicarea posturii se realizeaza cu frecventa de 500 Hz.