

Relatório da atividade 3

Nessa terceira atividade foi usado o modelo de carrinho anteriormente apresentado, portanto, o que foi preciso implementar nessa atividade, foi um código em python para que o carrinho andasse em linha reta.

Porém, antes de tudo, foi preciso instalar algumas bibliotecas para que o vscode conversasse com o Coppelia. As bibliotecas baixadas foram:

- `python3 -m pip install pyzmq cbor2`
- `pip install coppeliasim_zmqremoteapi_client`
- `pip install coppeliasim_zmqremoteapi_client==<version>`
- `pip install coppeliasim_zmqremoteapi_client==2.0.4`

Logo após as instalações, foi possível programar o carrinho, e ele conseguiu andar. A programação usada foi:

```
from coppeliasim_zmqremoteapi_client import RemoteAPIClient
import numpy as np
import time

client = RemoteAPIClient()
sim = client.getObject('sim')

print("Iniciando simulação...")
sim.startSimulation()

#motores
RodaD2 = sim.getObject('/direita')
RodaE2 = sim.getObject('/esquerda')
print("Motores obtidos com sucesso.")

L = 0.172 # distancia das rodas
r = 0.065 # raio das rodas

v = 0.2 # velocidade linear
w = 0 # velocidade angular

wr = ((2.0 * v) + (w * L)) / (2.0 * r)
wl = ((2.0 * v) - (w * L)) / (2.0 * r)

print(f"Velocidade angular da roda direita (wr): {wr}")
print(f"Velocidade angular da roda esquerda (wl): {wl}")

sim.setJointTargetVelocity(RodaD2, wr)
sim.setJointTargetVelocity(RodaE2, wl)
print("Velocidades definidas com sucesso.")
```

```
time.sleep(10)

print("Parando a simulac o...")
sim.stopSimulation()

print("Encerrando conex o...")
client.close()
print("Conex o encerrada.")
```