

Scénario initiale : Voir la hauteur minimale détecter par les capteurs de Pepper 18/10:

I. Connecter le PC avec Pepper :

1. `roscore` (initialise et coordonne les nœuds ROS).
2. `roslaunch naoqi_driver .launch noa_ip :=192.168.1.140
roscore_ip=192.168.1.100 network_interface :=ens1`

`ifconfig`

(Lancer le fichier, initialise les nœuds ROS nécessaires pour communiquer avec Pepper en utilisant `naoqi_driver`. Dans ce cas, on spécifie l'adresse IP de Pepper (`noa_ip`), l'adresse IP du PC (`roscore_ip`), et l'interface réseau utilisée (`network_interface`).

3. `conda activate pepper_nouveau_environnement`

(Activer l'environnement Conda) .

4. `roslaunch test.py`

(Une fois les nœuds lancés, on exécute le script Python qui contient les instructions pour interagir avec Pepper – move forward only) .
Commanding using terminal.

`chmod` make the file executable

connect to PC (`naoqi_driver` initialized) and then execute the python code.

II. Test : Recherche de faille de détection :

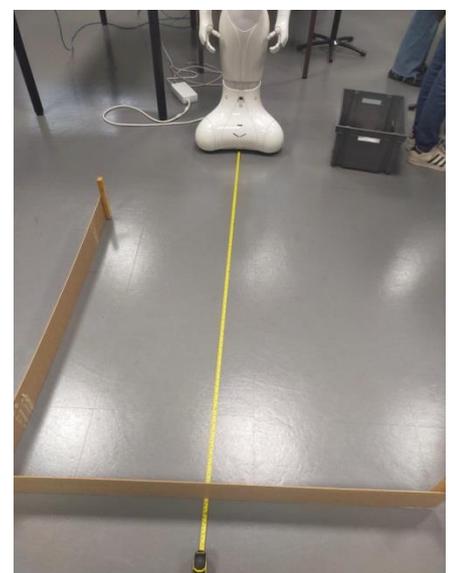
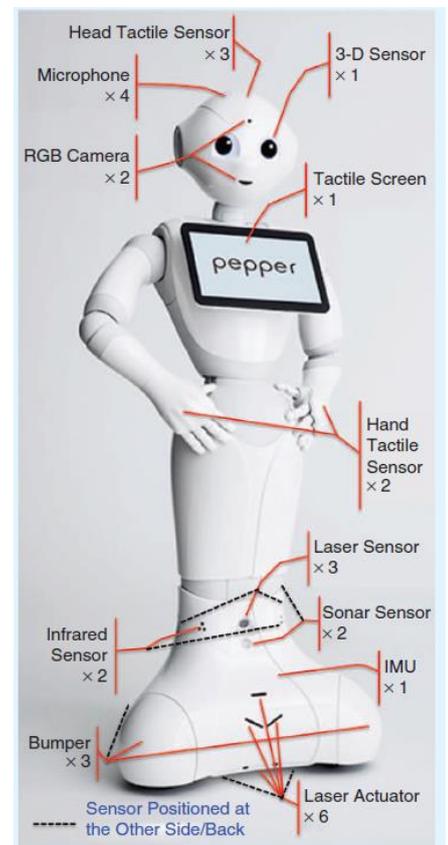
1. Capteurs Sonar/ cameras /Laser :
2. Choregraphe(Logiciel de programmation graphique).

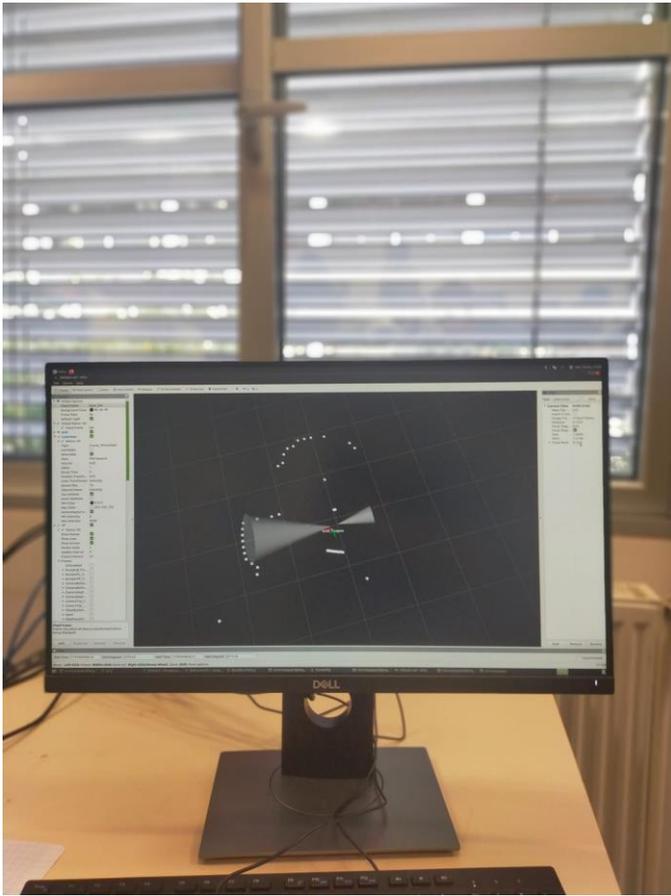
```
test.py  
move_forward()
```

III. RViz :

`RViz` outil de visualisation dans le système ROS utilisé pour afficher des informations issues de capteurs, des données de robot. Il permet de visualiser l'état d'un robot en temps réel et d'interagir avec des simulations ou des robots physiques.

- On fait des tests au capteur **Laser Actuator**.
- On veut déterminer la distance maximale pour laquelle on peut détecter un obstacle (Planche en ligne) sur `RVIZ`.





- On trouve qu'au delà de **1 m 80 cm** la ligne commence à se déformer.

IV. Observations :

Nous avons observé que la détection des obstacles à l'aide du capteur laser du robot Pepper dépend de la forme et de l'orientation de l'objet à détecter. Il est à noter que le capteur laser est puissant, capable de détecter des objets à des distances dépassant les 2 mètres..

