

1 Contrôle du bras Jaco dans le simulateur Gazebo par l'entremise de ROS

Cette section traite d'un environnement de travail hautement versatile permettant la représentation de plusieurs architectures de robots. Pour les besoins du rapport, seulement un exemple avec le bras Jaco sera discuté. De plus, la complexité de ROS et du simulateur Gazebo oblige de restreindre la présente section à la description sommaire des étapes nécessaires à la mise en marche de l'environnement et à l'utilisation de l'exemple fourni. Les descriptions fournies ne cherchent pas à être fondamentalement exactes, mais plutôt à expliquer de manière intuitive et succincte le fonctionnement de l'infrastructure à des lecteurs non-initiés. Il est important de noter que cet environnement est disponible seulement sur les plateformes Linux.

1.1 Description de l'architecture générale de ROS

ROS, *robot operating system*, peut être décrit comme un ensemble de programmes (*noeuds*) pouvant fonctionner de manière indépendante, entre lesquels des canaux de communications standardisés sont établis. Ces programmes communiquent entre eux en publiant l'information dans une liste des différents sujets (*topics*) publiés par les noeuds présentement en fonctionnement. Pour obtenir l'information publiée, un autre noeud peut s'abonner au sujet et ainsi lire l'information.

1.2 Description du simulateur Gazebo

Gazebo est un simulateur 3D pouvant fonctionner sans nécessiter l'installation de ROS. Cependant, il existe une version intégrée de Gazebo dans ROS Kinetic, la version de ROS utilisée pour Ubuntu 16.04. Gazebo peut être vu comme un noeud ROS qui publie des informations par rapport aux différents composants de la simulation. En utilisant le standard de fichier URDF, c'est dans ce programme que le robot est défini et que les contrôleurs des articulations sont créés. Suivant la création du robot, Gazebo publie ainsi les informations de l'état du robot et s'abonne à des topics permettant la réception de commandes de moteurs. La Figure 1 représente l'interface du simulateur Gazebo lorsque le robot Jaco est configuré.

La configuration du robot Jaco a été adaptée d'une suite de programmes (package) open-source publiée par [?].

1.3 Installation de ROS et de Gazebo

Comme mentionné plus haut ROS est disponible seulement sur les systèmes d'exploitation Linux. La séquence d'instructions suivante assume que le système d'exploitation Ubuntu 16.04 est fraîchement installé.

- Aller sur le site web suivant.
- Dans un invité de commande (terminal), entrer, dans l'ordre, les commandes aux étapes 1.2 et 1.3 du tutoriel.
- Dans un terminal, entrer la commande : `sudo apt-get install ros-kinetic-desktop-full`. Ceci installe ROS et Gazebo.
- Dans un terminal, entrer, dans l'ordre, les commandes à l'étape 1.5 du tutoriel.
- Dans un terminal, entrer la commande : `echo "source /opt/ros/kinetic/setup.bash" >> ~/.bashrc`
- Entrer ensuite la commande : `source ~/.bashrc`

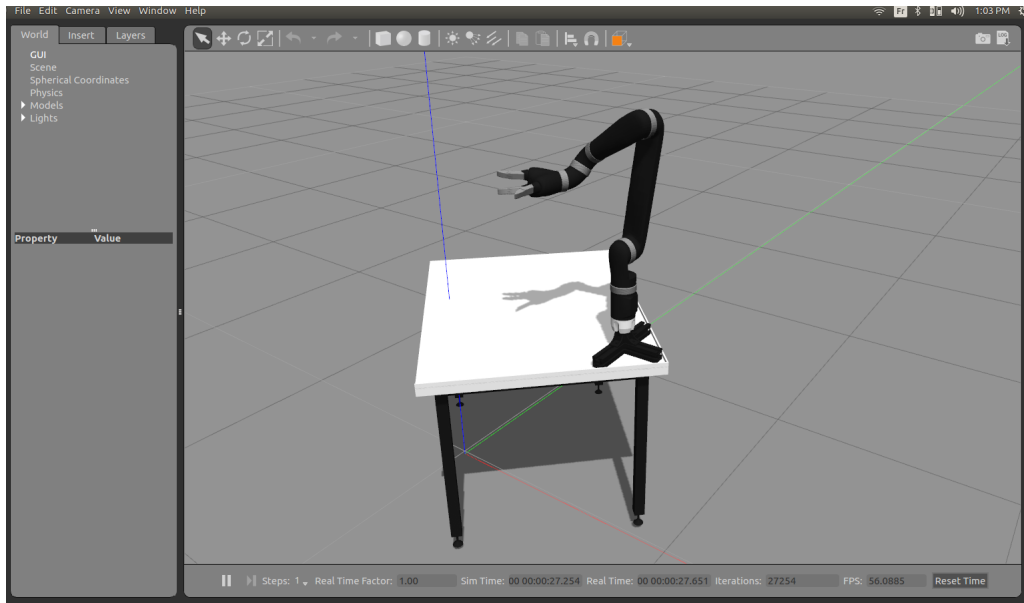


FIGURE 1 – Interface du simulateur Gazebo avec le robot Jaco configuré

- Dans un terminal, entrer la commande : `sudo apt-get install ros-kinetic-catkin`. Ce programme permet la construction d'un environnement de travail avec ROS.
- Aller sur le site web suivant.
- Entrer les commandes mentionnée dans le tutoriel sur cette page.
- Dans un terminal, entrer la commande : `echo "source ~/catkin_ws/devel/setup.bash" >> ~/.bashrc`
- Entrer ensuite la commande : `source ~/.bashrc`

Après l'exécution de ces étape, un environnement de travail pour ROS et Gazebo a été créé dans le fichier *catkin_ws* situé dans le répertoire *home* de votre plateforme Ubuntu. Les fichiers sur lesquels le projet repose seront placé dans le sous-fichier *src* de l'environnement de travail.