Práctica 2
Instalación de software para operar robots móviles

Laboratorio de Bio-Robótica

Construcción de Robots Móviles

Objetivos

- Familiarizar al alumno con el uso del software de control de versiones `git`.
- Aprender a utilizar el software desarrollado en el Laboratorio de Bio-Robótica para la operación de robots móviles autónomos.

1. Introducción

Git. Es un software de control de versiones *open source*, creado por Linus Torvalds. Está pensado para el desarrollo de software cuando éste posee una gran cantidad de código fuente y facilita el trabajo cooperativo mediante el manejo de *branching* y herramientas de solución de conflictos.

En este curso se usará `git` para el manejo de versiones del software necesario para operar los robots que se utilizarán en el desarrollo de prácticas y proyectos. Además, proporcionará al alumno herramientas para desarrollar de forma ordenada todos los códigos necesarios.

En la página `https://git-scm.com/` se pueden encontrar tutoriales para aprender más sobre `git`.

Justina. Es un robot de servicio desarrollado en el Laboratorio de Bio-Robótica de la Facultad de Ingeniería de la UNAM. En este curso se utilizarán algunos de los paquetes del software de Justina para familiarizar al alumno en el uso de Git y ROS.

2. Desarrollo

2.1. Instalación de Git y obtención del software

Nota. Se asume que el alumno ya tiene instalado Ubuntu 14.04 y ROS Indigo.

En una terminal, teclear los siguientes comandos:

```
$ sudo apt-get install git
$ cd
$ git clone https://github.com/mnegretev/RoboticsCourses.git
$ cd RoboticsCourses
```

Lo anterior descarga una copia del repositorio que contiene el material a usar durante el curso. Para descargar una actualización, teclee el siguiente comando:

```
$ git pull origin master
```

Puesto que en este momento no hay ninguna actualización disponible, se debe leer la siguiente salida:

```
From https://github.com/mnegretev/RoboticsCourses
 * branch            master     -> FETCH_HEAD
Already up-to-date.
```

2.2. Instalación de dependencias y compilación

Para poder compilar y ejecutar el software, es necesario instalar varias dependencias. Para ello, en una terminal teclee los siguientes comandos:

```
$ cd
$ cd RoboticsCourses/Prerequisites
$ ./Setup.sh
```

Esto comenzará a instalar varias bibliotecas como OpenCV (software para visión computacional), PCL (procesamiento de nubes de puntos), PrimeSense (controladores para cámaras RGB-D) y algunos paquetes necesarios de ROS. Cuando el script termine de ejecutarse, teclee los siguientes comandos para compilar el código:

```
$ cd catkin_ws
$ catkin_make
```

Una vez finalizada la compilación, ejecute los siguientes comandos:

```
$ source devel/setup.bash
$ roslaunch bring_up hardware_simul.launch
```

Se mostrará una ventana como la que se observa en la figura **??**. En otra terminal, ejecute el comando (es una sola línea, pero está en dos, por cuestiones de espacio):

```
$ rostopic pub -r 10 /hardware/mobile_base/speeds std_msgs/
  Float32MultiArray "{data:[0.02, 0.3]}"
```

El comando anterior *publica* el *tópico* `/hardware/mobile_base/speeds`, que acepta *mensajes* de tipo `Float32MultiArray`. El valor publicado es un arreglo de dos flotantes $[0.02, 0.3]$. El *nodo* `/hardware/mobile_base` está *suscrito* a este tópico y lo que hace es mover las llantas izquiera y derecha del robot con las velocidades correspondientes contenidas en el valor del tópico. Se debe observar que el robot comienza a dar vueltas.

2.3. Nodo para mover al robot

Escriba un nodo que publique en el tópico `/hardware/mobile_base/speeds` las velocidades necesarias para que el robot describa un cuadrado de 1m x 1m. El tópico debe publicarse 10 veces por segundo. El lenguaje puede ser C++ o Python.

3. Evaluación

Para que la práctica se condidere entregada se deben cumplir los siguientes puntos:

- El movimiento del robot en el visualizador debe describir claramente un cuadrado.
- El comando `rostopic echo` debe mostrar claramente que el tópico se está publicando.

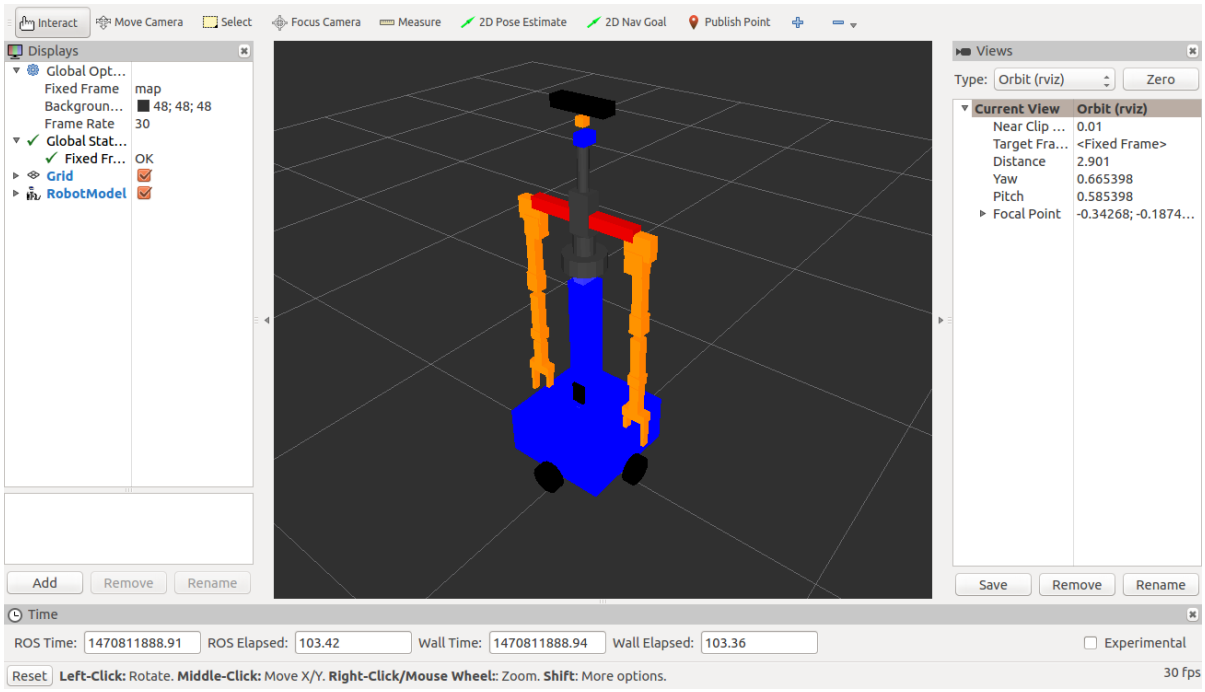

Figura 1: El visualizador `rviz`