

Práctica 2

Instalación de Software para Operar Robots Bípedos

Robots Bípedos Autónomos

Facultad de Ingeniería, UNAM

Objetivos

- Familiarizar al alumno con el uso del software de control de versiones `git`.
- Aprender a utilizar el software desarrollado para la operación de robots bípedos autónomos.
- Familiarizar al estudiante con el simulador Gazebo.

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 el robot que se utilizará 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`.

2. Desarrollo

2.1. Instalación de Git y obtención del software

Nota. Se asume que el alumno ya tiene instalado Ubuntu 16.04 y ROS kinetic.

En una terminal, teclear los siguientes comandos:

```
$ sudo apt-get install git
$ cd
$ git clone https://github.com/mnegretev/PAPIME_PE116219
$ cd PAPIME_PE116219
```

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 PAPIME_PE116219/SoftwareHumanoide
$ sudo ./Setup.sh
```

Esto comenzará a instalar varias bibliotecas y algunos paquetes de ROS necesarios. Cuando el script termine de ejecutarse, teclee los siguientes comandos para compilar el código:

```
$ cd
$ cd PAPIME_PE116219/SoftwareHumanoide
$ catkin_make -j2 -l2
```

Una vez finalizada la compilación, ejecute los siguientes comandos:

```
$ cd ~/PAPIME_PE116219/SoftwareHumanoide
$ source devel/setup.bash
$ export GAZEBO_MODEL_PATH=~/.PAPIME_PE116219/SoftwareHumanoide/src
  /hardware/nimbro_op_gazebo/models/${GAZEBO_MODEL_PATH}
$ roslaunch surge_et_ambula humanoid_simul.launch
```

Se mostrará una ventana como la que se observa en la figura 1. Presione el botón que se muestra encerrado en un cuadro rojo en la misma figura. Aparecerán otras dos ventanas como las que se muestran en las figuras 2 y 3.

En otra terminal, ejecute el comando y observe lo que sucede en el simulador Gazebo y en el visualizador RViz.

```
$ rostopic pub /hardware/head_goal_pose std_msgs/Float32MultiArray
  "{data:[0, 0.2]}"
```

El comando anterior *publica* el *tópico* `/hardware/head_goal_pose`, que acepta *mensajes* de tipo `Float32MultiArray`. El valor publicado es un arreglo de dos flotantes `[0.0, 0.2]`. El *nodo* `/hardware/control_remapper` está *suscrito* a este tópico y lo que hace es mover la cabeza del robot a un par de ángulos pan-tilt. El movimiento tilt positivo es hacia abajo para seguir la convención de la mano derecha en el sistema de referencia de la cabeza.

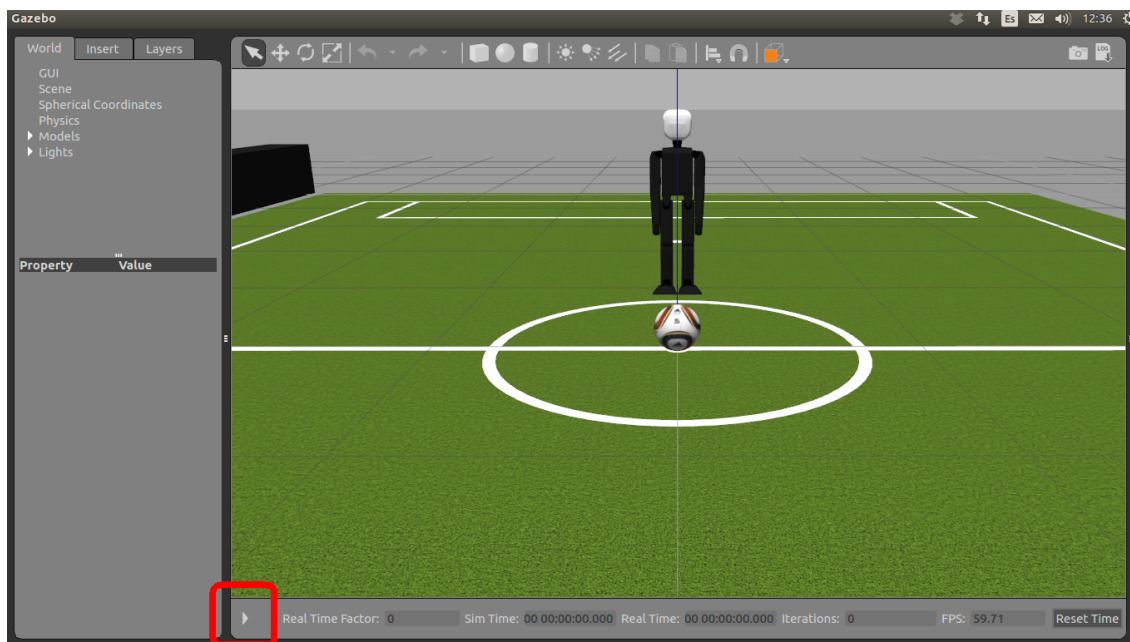


Figura 1: El simulador Gazebo.

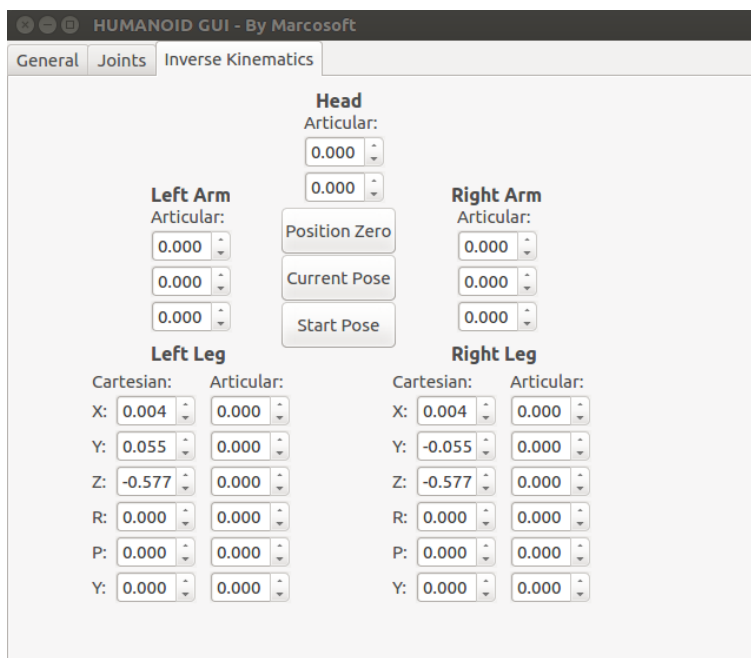


Figura 2: Interfaz gráfica para operación del robot bípedo.

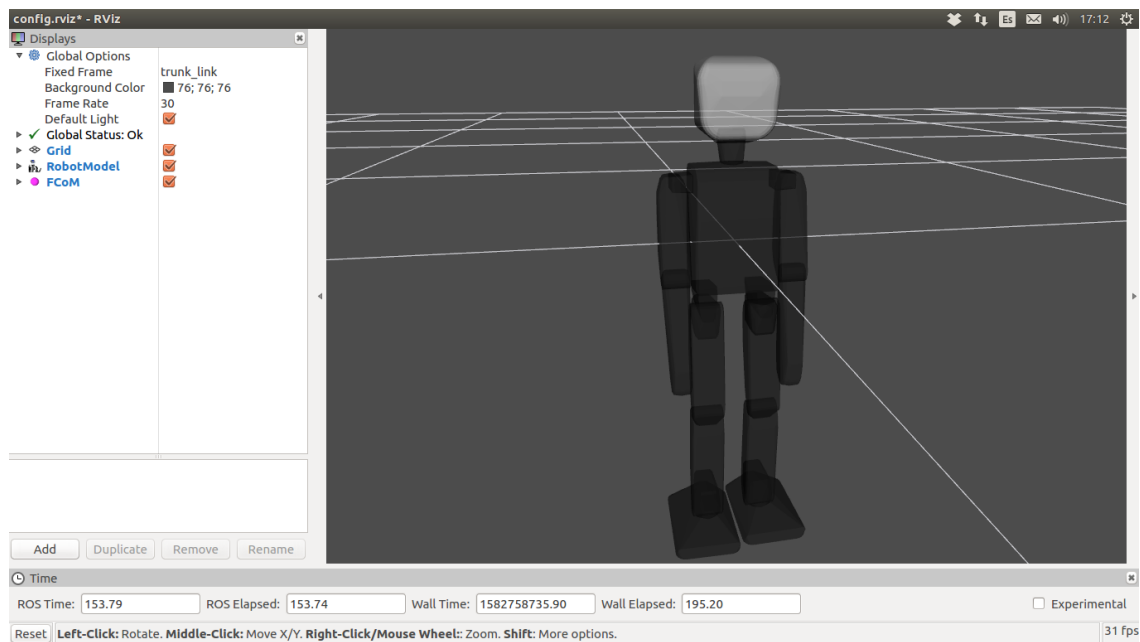


Figura 3: El visualizador RViz.

2.3. Nodo para mover la cabeza del robot

Escriba un nodo que publique en el t3pico `/hardware/head_goal_pose` los 3ngulos pan-tilt necesarios para que la cabeza del robot describa un gesto de “s3”, es decir, que mueva la cabeza de arriba a abajo. El t3pico debe publicarse 10 veces por segundo. El lenguaje puede ser C++ o Python.

3. Evaluaci3n

Para que la pr3ctica se considere entregada se deben cumplir los siguientes puntos:

- El movimiento del robot en el visualizador debe describir claramente el gesto “s3”.
- El comando `rostopic echo` debe mostrar que el t3pico se est3 publicando correctamente.