

**Curso:** Robótica IELE-3338

**Semestre:** 2020-20

**Profesor:** Carolina Higuera

**Asistentes:** Juan José García

**Monitores:** Andrés Felipe Florez y Paulina Acosta

**Publicación:** 13 de agosto de 2020

**Entrega:** 30 de agosto de 2020 - 11:59pm



## Taller 1 Introducción a ROS TurtleBot2

### Instrucciones

Resuelva cada uno de los puntos mostrados a continuación en los grupos inscritos en Sicua+. Cada grupo deberá entregar un documento de no más de 6 páginas que explique detalladamente la solución implementada para cada ejercicio. Es obligatorio que dicha descripción se realice usando herramientas de alto nivel tales como diagramas de bloques, diagramas de flujo y/o grafos de ROS entre otras (no se admite código en el documento). Para cada ejercicio también se espera que se presente un análisis y conclusión de la solución propuesta.

Además del documento, cada grupo deberá entregar una carpeta comprimida con el nombre *taller1\_grupo{x}.tar.gz* con los códigos utilizados para la solución de cada punto del taller. En particular, cada grupo deberá crear un paquete de ROS llamado *turtle\_bot\_x* donde *x* es el número del grupo. Dicho paquete deberá tener al menos un nodo de ROS por cada punto del taller. En la documentación se deben incluir instrucciones para ejecutar cada nodo y herramientas requeridas para su ejecución, así como también instrucciones de instalación de dichas herramientas. Proyectos que no tengan dicha documentación no se revisarán y su nota será equivalente a no haber entregado la solución del taller. Esto no aplicará si los comandos dados en la documentación no permiten ejecutar la solución.

El lenguaje de programación a utilizar es de libre decisión del grupo. Sin embargo, es necesario que los códigos tengan un mínimo de documentación la cual será tomada en cuenta en la clasificación. En todos los casos los grupos deben trabajar usando el entorno de simulación *V-REP* y bajo el framework de robótica *ROS*.

### 1 Entrega

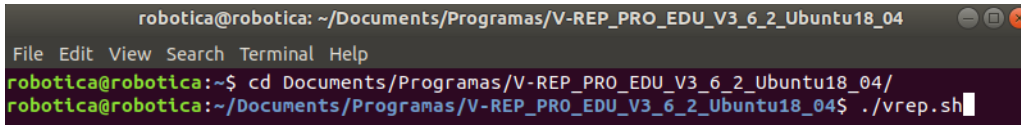
La entrega se realizará a través de Sicua+. Cada grupo deberá subir un único archivo comprimido con el nombre del paquete de ROS *turtle\_bot\_x* donde *x* es el número del grupo. La estructura de archivos del paquete debe ser la que tiene un paquete típico de ROS: carpetas *scripts*, *include*, *src*, *srv* (sólo las que sean necesarias) y archivos *README*, *CMakeLists.txt* y *package.xml*. Adicionalmente, el paquete deberá tener una carpeta *results* donde se encontrarán al menos el documento (pdf) de entrega, un video de demostración por cada punto de enunciado, gráficas de resultados y archivos. *.txt* de los recorridos guardados.

Entregas subidas a Sicua+ después de la fecha y hora máxima de entrega tendrán una penalización de 1.0 unidades en su calificación final por cada 15 minutos de retraso. Planee con anticipación el tiempo necesario para subir sus archivos de la entrega a Sicua.

## Entorno de simulación

V-Rep es un programa para simular robots que funciona con Linux. Es gratuito y de código abierto, mientras se use para fines no comerciales. V-Rep hace simulaciones realistas de cada una de las piezas que forman un robot. El usuario puede crear scripts para controlar el movimiento del robot y configurar sus sensores. V-Rep permite programar en Python, Lua, C++, Java y Matlab (entre otros lenguajes). Para este taller, se hará uso de la API para ROS que trae V-Rep.

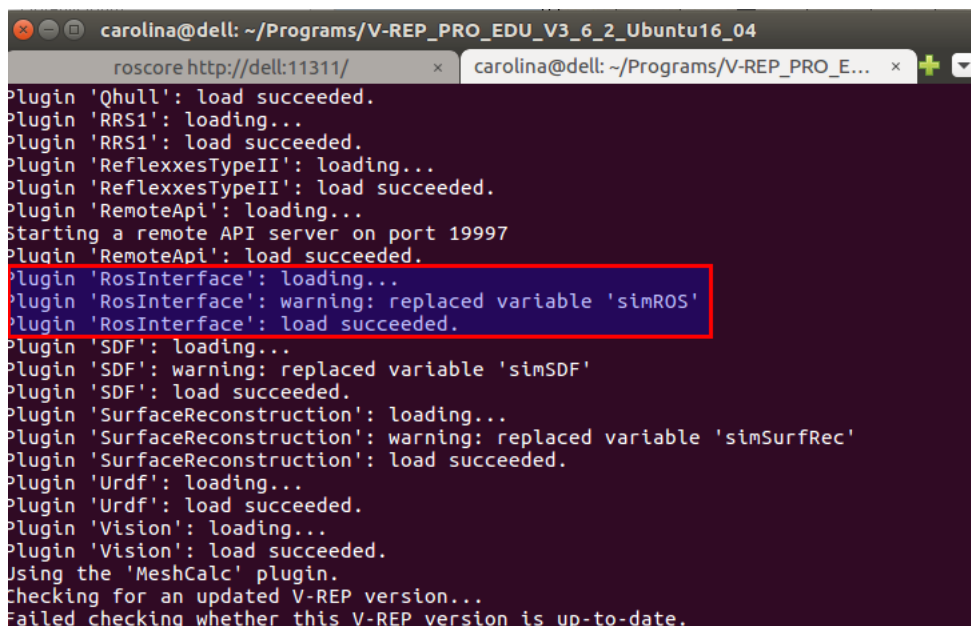
Una vez instalada la maquina virtual que se trabajará a lo largo del curso junto con el simulador y el framework de ROS se puede proseguir con el desarrollo de este taller. Para ejecutar el simulador, corra el *roscore* y en otra terminal, vaya a la carpeta en donde se encuentra el simulador (Ver video) y ejecute el script *vrep.sh* (ver Figura 1).



```
robotica@robotica: ~/Documents/Programas/V-REP_PRO_EDU_V3_6_2_Ubuntu18_04
File Edit View Search Terminal Help
robotica@robotica:~$ cd Documents/Programas/V-REP_PRO_EDU_V3_6_2_Ubuntu18_04/
robotica@robotica:~/Documents/Programas/V-REP_PRO_EDU_V3_6_2_Ubuntu18_04$ ./vrep.sh
```

Figure 1: Launch vrep.sh

Es importante revisar en la terminal que la interfaz para ROS sea correctamente cargada al lanzar V-Rep:



```
carolina@dell: ~/Programs/V-REP_PRO_EDU_V3_6_2_Ubuntu16_04
roscore http://dell:11311/
carolina@dell: ~/Programs/V-REP_PRO_E...
Plugin 'Qhull': load succeeded.
Plugin 'RRS1': loading...
Plugin 'RRS1': load succeeded.
Plugin 'ReflexxesTypeII': loading...
Plugin 'ReflexxesTypeII': load succeeded.
Plugin 'RemoteApi': loading...
Starting a remote API server on port 19997
Plugin 'RemoteApi': load succeeded.
Plugin 'RosInterface': loading...
Plugin 'RosInterface': warning: replaced variable 'simROS'
Plugin 'RosInterface': load succeeded.
Plugin 'SDF': loading...
Plugin 'SDF': warning: replaced variable 'simSDF'
Plugin 'SDF': load succeeded.
Plugin 'SurfaceReconstruction': loading...
Plugin 'SurfaceReconstruction': warning: replaced variable 'simSurfRec'
Plugin 'SurfaceReconstruction': load succeeded.
Plugin 'Urdf': loading...
Plugin 'Urdf': load succeeded.
Plugin 'Vision': loading...
Plugin 'Vision': load succeeded.
Using the 'MeshCalc' plugin.
Checking for an updated V-REP version...
Failed checking whether this V-REP version is up-to-date.
```

Figure 2: Ros Interface cargado correctamente

Luego, al ejecutar V-Rep, vaya a File→Open Scene y seleccione la escena que viene junto con esta guía *turtlebot2\_scene\_taller1.ttt*. Deberá ver la escena mostrada en la figura 3. La escena tiene un tamaño de 5m×5m.

En la esquina inferior derecha podrá ver el eje de coordenadas del marco de referencial global. Al iniciar la simulación, el robot inicia en la posición  $\xi_{I0} = [0, 0, 0]^T$  (Posición global respecto a la escena).

Al hacer doble clic en el modelo del robot (ver Figura 4) podrá ver la programación en Lua del Turtlebot2. En ese archivo podrá ver los tópicos que publica y a los que se suscribe el robot en la simulación.

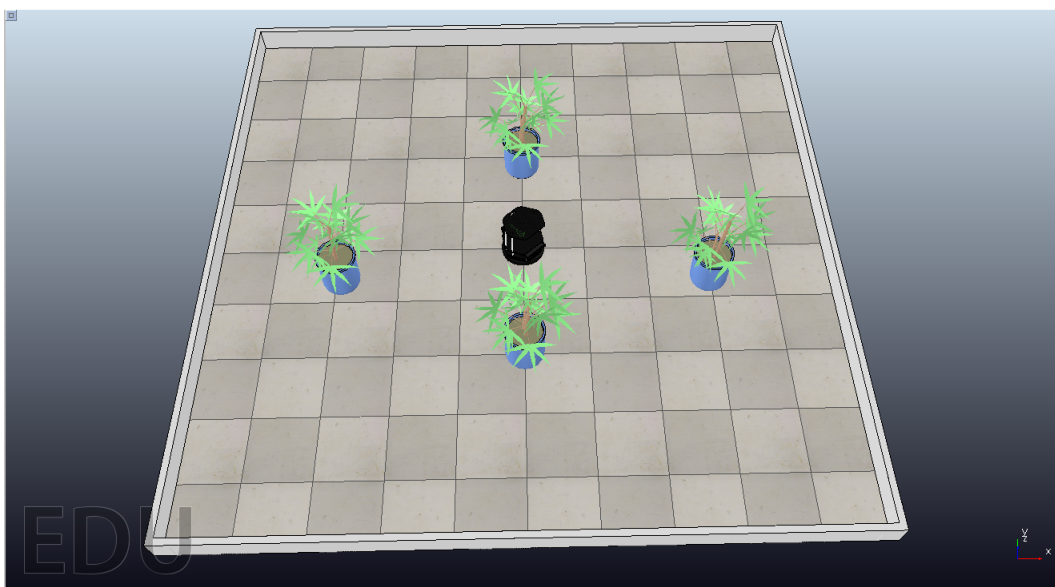


Figure 3: Escena en V-Rep turtlebot2\_scene.ttt

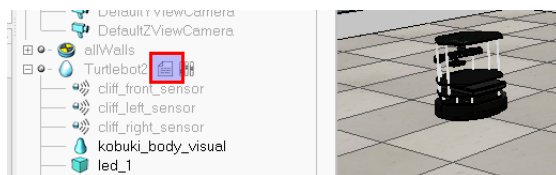


Figure 4: V-Rep .lua del modelo Turtlebot2

## Descripción de los tópicos

**/simulationTime:** tiempo de simulación en V-Rep

**/turtlebot\_position:** posición del Turtlebot2 en el marco inercial o global de referencia. El tipo de mensaje es `msg=geometry_msgs/Twist`:

- `msg.linear.x` = posición en x del robot en marco inercial
- `msg.linear.y` = posición en y del robot en marco inercial
- `msg.angular.z` = orientación del robot en marco inercial

**/turtlebot\_orientation:** orientación del robot, ángulos entre 0 y  $\pm\pi$  rad.

**/turtlebot\_cmdVel:** velocidad lineal y angular en el marco de referencial local del robot. El tipo de mensaje es `geometry_msgs/Twist`.

**/turtlebot\_wheelsVel:** velocidad lineal de la rueda izquierda y derecha del robot respectivamente. El tipo de mensaje es `std_msgs/Float32MultiArray`.

## Robot Turtlebot2

El Turtlebot2 es un robot diferencial montado sobre una base Kuboki. En la figura 5 se encuentra el drawing de la base.

El robot permite una velocidad lineal máxima de 70cm/s y rotacional de 180 deg/s.

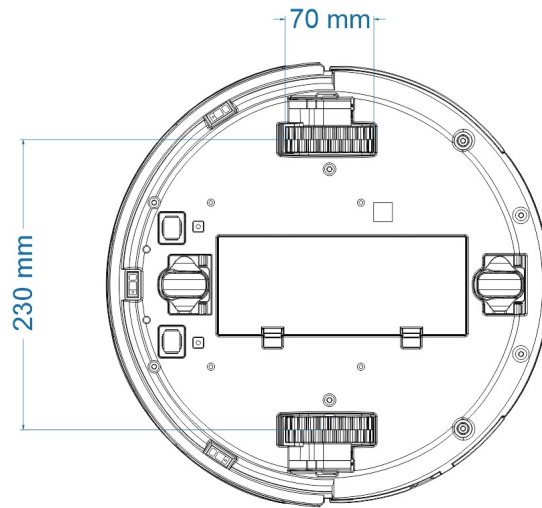


Figure 5: Base Kuboki

## Enunciado

1. **(1.5)** Cargue la escena `turtlebot2_scene_taller1.ttt` en V-Rep. Cree un nodo en ROS, llamado `/turtle_bot_teleop`, que permita a un usuario controlar por teclado el robot Turtlebot2 en V-Rep. Es decir, que las velocidades que son publicadas al robot de forma lineal y angular (en los tópicos respectivos) coincidan de forma natural con la distribución de teclas básicas (**Como mínimo el robot debe tener cuatro movimientos diferentes**). **Aclaración:** Si no se presiona ninguna tecla el robot se debe quedar quieto. El usuario debe definir el valor de la velocidad lineal y angular, en el marco de referencial local, con la que se mueve el robot al iniciar el nodo. No se admitirá tener que modificar el código para actualizar la velocidad. La velocidad lineal y angular del robot debe ser enviada a través del tópico `/turtlebot_cmdVel`.
2. **(1.5)** Cargue la escena `turtlebot2_scene_taller1.ttt` en V-Rep. Cree un nodo de ROS, llamado `/turtle_bot_position` que permita visualizar la posición en tiempo real del Turtlebot2 en V-Rep. Es decir, el programa deberá ir mostrando en la pantalla una gráfica donde se representa la posición del robot en el marco global de referencia en tiempo real y muestre el camino recorrido por el mismo desde donde se inició la simulación. Al finalizar la simulación en V-Rep, esta gráfica debe ser guardada en un archivo `trayectoria_punto2.png` en la carpeta **results** dentro del paquete.
3. **(0.5)** Complemente el nodo `/turtle_bot_teleop` para que al iniciar el nodo se le pregunte al usuario por terminal si desea guardar el recorrido del robot. En caso que la respuesta sea afirmativa, debe guardar en un archivo `.txt` la secuencia de acciones que realizó el usuario durante el recorrido del robot. El nombre del archivo debe ser preguntado al usuario y capturado desde la terminal.
4. **(1.5)** Cree un nodo de ROS, llamado `/turtle_bot_player`, que a partir de un archivo `.txt` de un recorrido guardado, reproduzca la secuencia de acciones del robot. La partida a reproducir debe ser solicitada a partir de un servicio y dicha llamada al servicio puede ser realizada directamente desde terminal con el comando `rosservice call`. En este caso no hay intervención del usuario. El nodo `/turtle_bot_player` debe proveer el servicio de recibir el nombre de la partida a reproducir.

**Nota:** Solo se debe replicar el recorrido del robot, no todas las acciones del usuario.

**Nota:** Es importante que para el desarrollo de este taller, tenga en cuenta el concepto de *Threads*. Puede guiarse por algunos ejemplos, a continuación se presenta uno de ellos.

<https://realpython.com/intro-to-python-threading/>

## Calificación

Cada punto de la sección Enunciado se calificará de acuerdo a la siguiente rúbrica. La calificación de cada punto será:

$$cal\_punto = \frac{valor\_punto * puntaje\_rubrica}{100}$$

Descripción solución	0 pts. El grupo no entrega el documento o este no describe en alto nivel la solución planteada.	5 pts. El grupo describe en el documento de entrega la solución diseñada en alto nivel (gráficas, diagramas de flujo, de bloques, etc), sin embargo esta descripción no es clara.	10pts. El grupo describe en el documento de entrega de manera clara y utilizando herramientas apropiadas la solución diseñada en alto nivel (gráficas, diagramas de flujo, de bloques, etc).
	0 pts. No muestra el grafo de ROS con la relación entre nodos y tópicos	2 pts. Se muestra el grafo de ROS con la relación entre nodos y tópicos pero no es explicado	5pts. Se muestra el grafo de ROS con la relación entre nodos y tópicos y éste es explicado con relación a la solución descrita
Implementación de la solución	0 pts. El grupo no describe en el documento detalles de implementación de la solución planteada o los descritos no son suficientes para entender las herramientas utilizadas que permitieron solucionar el taller.	5 pts. El grupo lista en el documento de entrega algunos detalles de implementación de la solución planteada, más estas no son descritas, es decir, no se explica su propósito o razón de estar en la solución.	10 pts. El grupo describe en el documento de entrega detalles de implementación de la solución planteada, tales como herramientas software (externas) utilizadas, tiempos de muestreo, elementos para la toma de decisiones, funciones especiales usadas en la solución y cualquier otro elemento que sea importante dentro de la misma
Estructura entrega	0 pts. El grupo no entrega su solución con todos los requerimientos técnicos descritos en las instrucciones.		2 pts. El grupo entrega su solución con los requerimientos técnicos descritos en las instrucciones.
	0 pts. Los puntos no están correctamente estructurados según los proyectos típicos de ROS		2 pts. Cada punto está correctamente estructurado según los proyectos típicos de ROS
	0 pts. Todos o alguno de los archivos fuente (con código) no tienen comentarios con una mínima descripción sobre qué hace cada parte del programa		2 pts. Todos los archivos fuente (con código) tienen comentarios con una mínima descripción sobre qué hace cada parte del programa
	0 pts. No implementa toda la solución dentro de un mismo paquete de ROS con el número solicitado/apropiado de nodos		2 pts. Cada punto está implementado dentro de un mismo paquete de ROS con el número solicitado/apropiado de nodos

	0 pts. No hay un archivo de documentación que explica cómo se ejecuta cada nodo del paquete, qué dependencias adicionales tiene y cómo se instalan dichas dependencias		2 pts. Hay un archivo de documentación que explica cómo se ejecuta cada nodo del paquete, qué dependencias adicionales tiene y cómo se instalan dichas dependencias
Funcionamiento	0 pts. El grupo no envía un video de demostración por cada punto del enunciado.	5 pts. El grupo envía un video de demostración por cada punto del enunciado, sin embargo no es claro la secuencia de comandos para ejecutar la solución o no se muestran todos los resultados.	10 pts. El grupo envía un video de demostración por cada punto del enunciado. En el video se debe ver explícito desde que se abre la primera terminal para comenzar la ejecución del proyecto. Se deben ver la secuencia de comandos usados para ejecutar la solución, así como todos los resultados que se deben mostrar por pantalla.
	0 pts. La solución no cumple con ninguno de los requerimientos de funcionamiento del enunciado	7 pts. La solución cumple con algunos de los requerimientos de funcionamiento del enunciado	15 pts. La solución cumple con todos los requerimientos de funcionamiento del enunciado
	0 pts. No posible replicar los resultados descritos por el grupo	5 pts. Es posible replicar los resultados descritos por el grupo pero estos no coinciden con los resultados mostrados en el video	10 pts. Es posible replicar los resultados descritos por el grupo y éstos coinciden
Resultados y análisis	0 pts. El grupo no presenta en el documento de entrega los resultados obtenidos en cada punto usando gráficas y fotos (entre otras) .	7 pts. El grupo describe en el documento de entrega los resultados obtenidos en cada punto usando gráficas y fotos (entre otras) pero estas no demuestran el funcionamiento esperado.	15 pts. El grupo describe en el documento de entrega los resultados obtenidos en cada punto usando gráficas y fotos (entre otras) que demuestran el funcionamiento esperado.
	0 pts. Las figuras no son comentadas y no se hace un análisis de dichos resultados	7 pts. Todas las figuras son comentadas pero el análisis de dichos resultados no justifica o no es coherente con los resultados mostrados en el video.	15 pts. Todas las figuras son comentadas y se hace un análisis de dichos resultados