

Curso: Robótica IELE-3338

Semestre: 2020-20

Profesor: Carolina Higuera

Asistentes: Juan José García

Monitores: Andrés Felipe Florez y Paulina Acosta

Publicación: 4 de septiembre de 2020

Entrega: 18 de septiembre de 2020 - 11:59pm



Taller 2 Cinemática de robot diferencial 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 los códigos utilizados para la solución de cada punto del taller. En particular, cada grupo deberá crear un paquete de ROS llamado **taller2_x** donde **x** es el número del grupo. Dicho paquete deberá tener 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 también 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 tenida en cuenta en la calificación.

Entrega

La entrega se realizará a través de Sicua+. Cada grupo deberá subir un único archivo comprimido (tar.gz) con el paquete de ROS **taller2_x**. La estructura de archivos del paquete debe ser la que tiene un paquete típico de ROS: carpetas **scripts**, **src**, **include** (sólo las que sean necesarias) y archivos **README**, **CMakeList.txt** y **package.xml**. Adicionalmente, el paquete debe tener una carpeta **results** donde se encontrará al menos el documento pdf de entrega y los videos de resultados. Los ejercicios que no requiera desarrollar programas, se evaluarán únicamente con la descripción de la solución.

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.

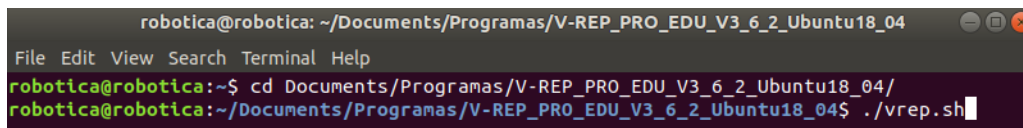
Aclaraciones

Recuerden que la asistencia a la sesión de laboratorio por parte de todos los miembros del grupo es obligatoria. Esto con el objetivo de llevar un registro y conocer el estado de avance y trabajo sobre el taller. Ausencias superiores al 20% de las clases de laboratorio correspondientes a cada estudiante sin justificación, equivale a una penalización de 2.0 unidades en la calificación final individual de la nota del taller.

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.

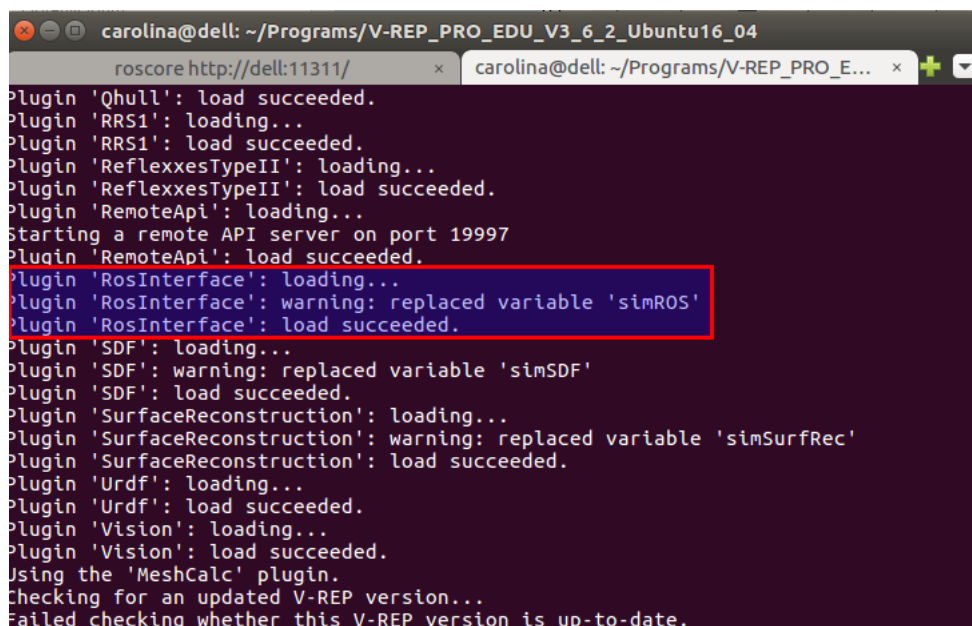
Una vez instalada la maquina virtual que se trabajará a lo largo del curso junto con el simulador y el framework de ROS se puede proseguir con el desarrollo de este taller. Para ejecutar el simulador, corra el *roscore* y en otra terminal, vaya a la carpeta en donde se encuentra el simulador (Ver video) 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 1997
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 $5m \times 5m$.

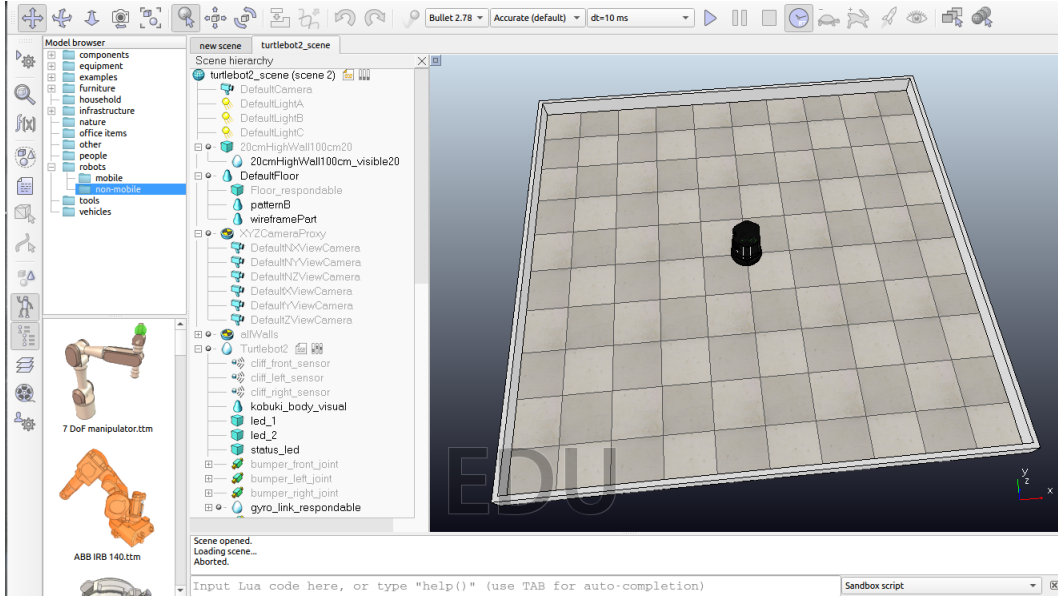


Figure 3: Escena en V-Rep turtlebot2_scene.ttt

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.

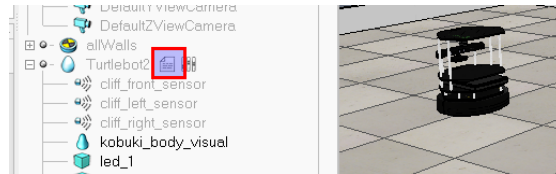


Figure 4: V-Rep .lua del modelo Turtlebot2

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.

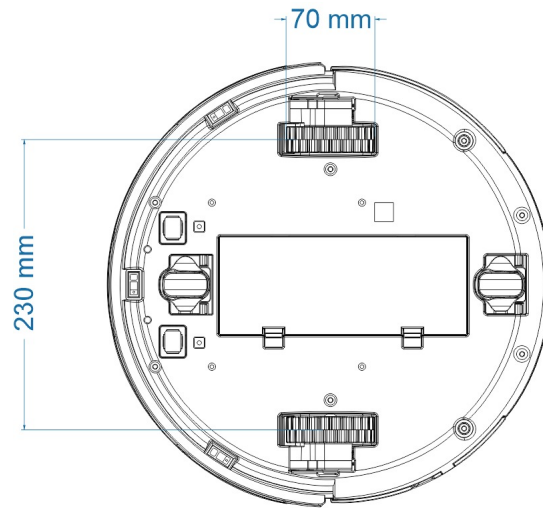


Figure 5: Base Kuboki

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

Enunciado

1. **(1.5)** Un robot está ubicado en $(x, y, \theta)_I = (1m, 0.5m, \pi/4)_I$ en el marco inercial. El sensor láser está montado encima del robot en $(0.2m, 0m)_R$ con respecto al marco de referencia local del robot. Las mediciones de distancia del láser se encuentran en el archivo `laserscan.dat`, el cual se encuentra anexo. La primera medición se toma a un ángulo de $\alpha = -\pi/2$ (en el marco de referencia del sensor) y la última a un ángulo $\alpha = +\pi/2$. Todas las mediciones se toman a distancias angulares iguales (todos los ángulos en radianes).

Pueden cargar los datos y el ángulo correspondiente a cada medición, así:

```
laserdata = np.loadtxt( 'laserscan.dat' )  
theta = np.linspace(angle_min, angle_max, num_data)
```

- (a) (0.3) Grafique todos los puntos finales del láser en el marco de referencia del sensor.
 - (b) (0.5) Al graficar, es posible notar que algunas mediciones (puntos) inesperados. Sugiera una explicación a lo que está sucediendo, basandose en la gráfica del numeral a).
 - (c) (0.7) Aplique transformaciones homogéneas para graficar los puntos del láser en el marco de referencia global. Grafique con un x la posición $(x, y)_I$ del robot y con una x la posición $(x, y)_I$ del sensor láser.
2. **(1.5)** Cargue la escena `turtlebot2_scene.ttt` en V-Rep. Cree un nodo en ROS que lea de un archivo de texto un perfil de velocidad para el robot Turtlebot2 de V-Rep y simule su movimiento en el marco de referencia global de acuerdo al perfil de velocidad dado. El nombre del archivo se debe pasar como argumento cuando se lanza el nodo. El programa debe mostrar por pantalla una gráfica en el marco global de referencia que vaya mostrando la posición actual del robot (calculada por odometría) y el camino recorrido. En la misma gráfica, debe mostrar la posición y el camino recorrido según la información enviada desde el simulador por el tópico `/turtlebot_position`. Al finalizar el movimiento del robot, se debe desplegar una ventana en la que se observe el error de posición en el eje vertical y el tiempo de simulación en el eje horizontal.

El cálculo de la cinemática del robot debe ser implementado por cada grupo: no se admite utilizar programas de terceros. Tanto la gráfica generada del error como la de la trayectoria realizada por el robot deben ser almacenadas en archivos `trayectoria_punto3.png` y `error_punto3.png` en la carpeta **results** dentro del paquete al finalizar la simulación.

El archivo de entrada se debe almacenar en una carpeta **resources** dentro del paquete y su estructura será de la siguiente manera: la primera línea tendrá un número entero que corresponderá al número de velocidades n que deberán ser simuladas por el robot. Las siguientes n líneas tendrán 3 números decimales separados por espacio que corresponderán a la velocidad del motor derecho, motor izquierdo y el tiempo que se mantendrán dichas velocidades. El tiempo de simulación específico debe ser leído de V-Rep e integrado en el modelo cinemático. La velocidad de las ruedas debe ser enviada a través del tópico `/turtlebot_wheelsVel`. En el informe escrito incluya las ecuaciones de la cinemática usadas en el programa junto con el valor de los parámetros. Realice y entregue un vídeo `punto3.mp4` dentro de la carpeta **results** mostrando cómo se mueve el robot en la simulación según el perfil dado.

3. **(2.0)** Cargue la escena `turtlebot2_scene_taller2.ttt` en V-Rep. Cree un nodo en ROS que reciba como parámetros de entrada al lanzar el nodo, 3 números que corresponderán a la posición final del robot $\xi_{If} = [x_f, y_f, \theta_f]^T$ e implemente la cinemática y la ley de control necesarios para llevar el robot desde la posición inicial a la posición final. Asegúrese que el robot en V-Rep inicia la simulación en la posición $\xi_{I0} = [0, 0, -\pi]^T$. Programe su nodo para que si el usuario no ingresa parámetros, se asuma que la posición final es $\xi_{If} = [-2, -2, -3\pi/4]^T$.



Figure 6: Ejemplo de figura delfin

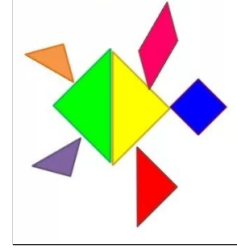


Figure 7: Ejemplo de figura tortuga

Al ejecutar el nodo se debe mostrar en tiempo real una gráfica, en el marco global de referencia, con la posición estimada del robot y la posición del robot según el simulador, además del camino recorrido en los dos casos. Al finalizar el movimiento del robot, se debe desplegar una ventana en la que se observe el error de posición del robot en V-Rep en el eje vertical y el tiempo de simulación en el eje horizontal. El cálculo de la cinemática del robot y de la ley de control deben ser implementados por cada grupo; no se admite usar programas realizados por terceros. Tanto la gráfica generada del error como la de la trayectoria realizada por el robot deben ser guardadas en archivos `error_punto4.png` y `trayectoria_punto.png` en la carpeta **results** dentro del paquete al finalizar la simulación. El tiempo de simulación debe ser leído de V-Rep. En el informe escrito incluya las ecuaciones de la cinemática y control usadas en el programa junto con el valor de los parámetros. Especifique cómo se escogieron los valores de k_ρ ; k_α y k_β . Realice y entregue un vídeo `punto3.mp4` dentro de la carpeta **results** mostrando cómo se mueve el robot en la simulación.

4. **Bono +2.0** Cargue la escena `turtlebot2_scene_taller2.ttt` en V-Rep. Modifique el modelo del robot agregándole un lápiz de color rojo disponible en la carpeta `components/modifiers/felt_pen` de forma que este sea capaz de escribir o dejar una marca al moverse. Cree un nodo en ROS con el que a partir del controlador desarrollado en el punto 3, el robot pueda llevar a cabo una trayectoria similar a la que se muestra en cualquiera de las figuras 6 y 7 (**Delfin y Tortuga**). La única restricción es que la figura debe encontrarse en un área de 2×2 m. Explique cómo establece la trayectoria que debe seguir el robot. Realice y entregue un vídeo `bono.mp4` dentro de la carpeta **results** mostrando cómo se mueve el robot en la simulación.

El bono solo aplica si la solución entregada está completa y funciona correctamente. No tiene calificaciones parciales: 0.0 o 2.0

Calificación

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

$$cal_punto = \frac{valor_punto * puntaje_rubrica}{100} + bono$$

| | | | |
|-------------------------------|--|---|---|
| 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 |