

Curso: Robótica IELE-3338

Semestre: 2022-10

Profesor: Juan José García

Asistentes: Susana Marcela Chavez

Monitores: Vilma Tirado

Publicación: 12 de febrero de 2022

Entrega: 13 de marzo de 2022 - 11:59pm



Taller 2

Diseño de un robot diferencial

Instrucciones

Diseñe un robot diferencial u omnidireccional que permita moverse y comunicarse a través de ROS. Su diseño debe contener planos mecánicos y electrónicos como también las medidas generales de su robot. El robot debe ser de 20x25 cm. A continuación se listan algunos materiales que pueden ser útiles para su diseño:

Materiales propuestos

1. Interruptor de encendido/apagado Costo promedio: \$ 2.500,00
2. Rueda castor (rueda loca) Costo promedio: \$ 5.000,00
3. Servo giro continuo Costo promedio: \$ 20.000,00
4. Ruedas de carrito Costo promedio: \$ 20.000,00 Link <https://www.homecenter.com.co/homecenter-co/product/399958/rueda-diablo-de-hule-de-5-pulg-con-rhin-balero-10mm/399958/?kid=bnnext1031763&shop=googleShopping>
5. Puente H - Referencia L293D (Dirección del motor)

Resuelva cada uno de los puntos mostrados a continuación en su robot diseñado y ensamblado en los grupos inscritos en Bloque Neón. Cada grupo deberá entregar un documento de no más de 10 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. En dicho documento también deberá agregar los planos mecánicos, electrónicos, la lista de materiales y explicar la integración de sus sistemas.

Además del documento, cada grupo deberá entregar una carpeta comprimida con el nombre *taller2_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 *mi_robot_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 calificación. En todos los casos los grupos deben trabajar bajo el framework de robótica *ROS*.

1 Entrega

La entrega se realizará a través de Bloque Neón. Cada grupo deberá subir un único archivo comprimido con el nombre del paquete de ROS `mi_robot_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, gráficas de resultados y archivos .txt de los recorridos guardados.

Entregas subidas a Bloque Neón 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.

La sustentación se realizará en las secciones de laboratorio de manera presencial en la semana del 14 al 18 de marzo del 2022. Allí demostrarán cada uno de los puntos del enunciado.

Descripción de los tópicos sugeridos a crear

/robot_position: posición del robot 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

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

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

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

Enunciado

1. Cree un nodo en ROS, llamado `/robot_teleop`, que permita a un usuario controlar por teclado su robot. 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 `/robot_cmdVel`.
2. Cree un nodo de ROS, llamado `/robot_interface` que permita visualizar la posición en tiempo real de su robot, a través de una interfaz gráfica. 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 inició. La interfaz debe contar con el espacio para asignarle un nombre a la gráfica y poderlo guardar en el directorio deseado, al finalizar el recorrido.
3. Complemente el nodo `/robot_teleop` y el nodo `/robot_interface` para que al iniciar el nodo se le pregunte al usuario en la interfaz 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 interfaz.
4. Cree un nodo de ROS, llamado `/robot_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 ser realizada directamente desde la interfaz creada anteriormente. El nodo `/robot_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

*Lista materiales (4), plano mecánico (6), eléctrico/electrónico(6), descripción funcionamiento del robot detallada (8), integración detallada de la parte eléctrica y mecánica (6). *30 explicación planos y demás.

*20 explicación nodos en informe.

*50 sustentación.

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

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

Diseño del robot	0 pts. El grupo no entrega la lista de materiales.		4pts. El grupo entrega la lista de materiales donde cada componente tiene su nombre, referencia y link de referencia (si aplica)
	0 pts. El grupo no entrega el plano mecánico.		6pts. El grupo entrega el plano mecánico utilizado en el diseño del robot de forma clara y organizada utilizando las herramientas apropiadas.
	0 pts. El grupo no entrega el plano electrónico.		6pts. El grupo entrega el plano electrónico utilizado en el diseño del robot de forma clara y organizada utilizando las herramientas apropiadas.
	0 pts. El grupo no explica la integración entre la parte mecánica y la parte electrónica.		6pts. El grupo explica de forma clara y organizada la integración entre la parte mecánica y electrónica utilizando las herramientas apropiadas.
	0 pts. El grupo no explica el funcionamiento general del robot diseñado.		8pts. El grupo explica de forma clara y detallada el funcionamiento general de su robot diseñado
Solución	0 pts. El grupo no crea los nodos para desarrollar cada punto del enunciado.		6 pts. El grupo crea los nodos necesarios para desarrollar cada punto del taller. Se puede ver la secuencia de comandos usados para ejecutar la solución.
	0 pts. La solución no cumple con ninguno de los requerimientos de funcionamiento del enunciado	6 pts. La solución cumple con algunos de los requerimientos de funcionamiento del enunciado	8 pts. La solución cumple con todos los requerimientos de funcionamiento del enunciado.

	0 pts. El grupo no presenta en el documento de entrega los resultados obtenidos en cada punto usando gráficas y fotos (entre otras) .	4 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.	6 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.
Sustentación	0 pts. El grupo no presenta sustentación de este punto.		15 pts. El grupo puede controlar por teclado su robot. Este tiene al menos cuatro movimientos y se queda quieto cuando ninguna tecla es presionada
	0 pts. El grupo no presenta sustentación de este punto.		15 pts. Es posible visualizar la posición en tiempo real del robot a través de una interfaz gráfica. El programa va mostrando en la pantalla una gráfica donde se representa la posición del robot en el marco global de referencia en tiempo real y muestra el camino recorrido por el mismo desde donde inició.
	0 pts. El grupo no presenta sustentación de este punto.		5pts. Al iniciar el nodo se le pregunta al usuario en la interfaz si desea guardar el recorrido del robot. En caso que la respuesta sea afirmativa, se guarda en un archivo .txt la secuencia de acciones que realizó el usuario durante el recorrido del robot. El nombre del archivo es al usuario y capturado desde la interfaz.
	0 pts. El grupo no presenta sustentación de este punto.		15pts. El grupo crea un nodo 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 ser realizada directamente desde la interfaz creada anteriormente.