

Curso: Robótica IELE-3338
Semestre: 2019-10
Profesor: Carlos A. Quintero Peña
Publicación: 2 de abril

ROBÓTICA: PROYECTO DE APLICACIÓN

1. Motivación

Los robots móviles han crecido significativamente en la cantidad de aplicaciones comerciales y de servicio y esta tendencia sólo tiende a aumentar con el desarrollo de la tecnología. El desarrollo de un robot móvil para casi todas las aplicaciones comprende la integración de una gran cantidad de disciplinas, haciéndola un área tan interdisciplinaria como es posible. Típicamente, un robot móvil tiene lo que se conoce como una “capa cognitiva”, la cual consiste en las capacidades del robot para desarrollar una lógica y comportamientos que le permiten llevar a cabo la tarea para la cual fue diseñado. Esta capacidad cognitiva depende, en general, de la aplicación para la cual fue creado el robot, sin embargo, todos los robots móviles comparten una porción de dicha capa que consiste en su habilidad para movilizarse de manera robusta.

La ejecución de este proyecto tiene dos grandes objetivos principales: Por un lado, pretende reforzar en los estudiantes los conceptos básicos de robótica móvil vistos en el curso a través de una experiencia práctica, permitiéndoles contrastar dichos conceptos y el comportamiento en simuladores con un sistema real. Por otro lado, se espera que los estudiantes se enfrenten a problemas y situaciones que sólo surgen en la construcción e implementación de un sistema complejo como un robot móvil y que sean capaces de usar los conceptos aprendidos a lo largo de su carrera para tomar decisiones de diseño e implementación.

Un aspecto fundamental para el éxito del proyecto está en el trabajo en equipo. Es importante que cada equipo conozca las fortalezas de sus integrantes y las aproveche para lograr el mejor resultado posible del mismo. Se espera que este proyecto genere un reto en cada estudiante que les permita crear un producto de calidad, usando las habilidades adquiridas durante su formación alrededor de la robótica móvil, para lograr desarrollar un prototipo funcional.

2. Descripción General

Cada grupo deberá diseñar, construir y poner a prueba un robot móvil diferencial programado y controlado usando *Robot Operating System (ROS)* para que sea capaz de cumplir una tarea en particular dentro de un ambiente estructurado.

El reto de cada equipo consistirá en construir un robot que cumpla con los requerimientos dados, en el tiempo disponible y que esté dotado con las capacidades para resolver el problema planteado. Para ello deberán tomar decisiones de diseño y llevar a cabo una serie de etapas que les permitan lograr el objetivo de manera adecuada.

3. Especificaciones

En esta sección se describen las especificaciones de cada una de las partes del reto.

3.1. El Robot

Cada equipo deberá diseñar y construir (ensamblar) su propio robot. A continuación se detallan algunas restricciones que éste debe cumplir:

- El robot deberá contar con dos ruedas estándar fijas y tracción diferencial. Es decisión de cada equipo incluir o no en su diseño *ruedas locas* adicionales
- El tamaño máximo del robot es $250mm \times 200mm$. No hay restricción con respecto su altura
- El robot deberá alimentarse a través de baterías. No se permitirá la realización de pruebas con cables que salgan del mismo
- La electrónica necesaria para el control del robot deberá estar en tarjetas de circuito impreso (no se aceptarán circuitos en Protoboard)
- El robot deberá contar con un sistema de información para el humano que le permita comunicar de una manera muy sencilla en qué estado se encuentra (LEDs, parlantes, etc). Esta comunicación debe servir para que un operario pueda conocer estados básicos del robot en la prueba **simplemente viéndolo/escuchándolo actuar**
- El robot debe contar con un sistema electrónico digital para la lectura de la información de los sensores y para las señales de control hacia los actuadores. Es un requisito que dicho sistema digital tenga instalada alguna versión de ROS, de manera que *toda la adquisición, procesamiento y generación de señales de control se haga desde allí*. Si algún equipo tiene la necesidad de ejecutar alguno de los procesos en un programa que no haga parte de ROS deberá solicitar permiso y justificarlo ante el profesor y los asistentes del curso. De la misma manera, si algún equipo desea utilizar recursos computacionales externos al robot (PC y otros), deberá solicitar el permiso
- Además de tener ROS, el sistema digital de adquisición y control del robot deberá contar con la posibilidad de conexión Wi-Fi
- El robot deberá contar con un sistema de odometría que le permita tener un estimado del movimiento de cada una de las ruedas (encoders)

3.2. Operación

La Figura 1 muestra la arquitectura general que deberá tener el sistema para poder realizar las pruebas. El robot diseñado por cada equipo deberá contar con ROS y Wi-Fi de manera que pueda establecer una comunicación TCP-IP a través de una red local con un computador externo (llamado el *master*). El *master* ejecutará un nodo de ROS llamado *master_node* y se encontrará en la red local conectado por Ethernet al *router*.

El *master* será el encargado de ejecutar *roscore*. Esto significa que cada robot no deberá ejecutar su propio *roscore* sino que se conectará al programa del *master*. Los robots diseñados por los equipos no deberán tener ningún botón de inicio ni deberá ser necesario acceder a su sistema operativo (usando

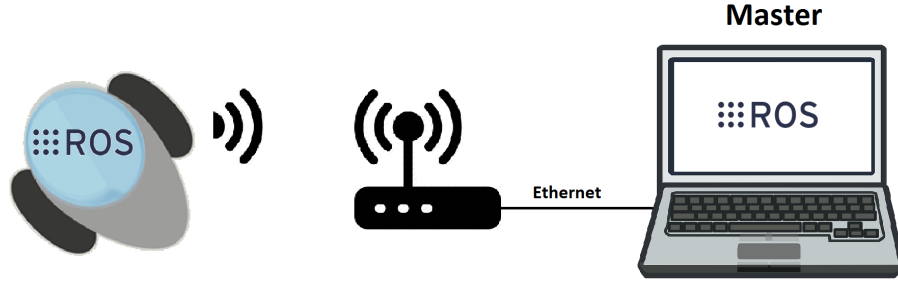


Figura 1: Arquitectura general de operación para las pruebas

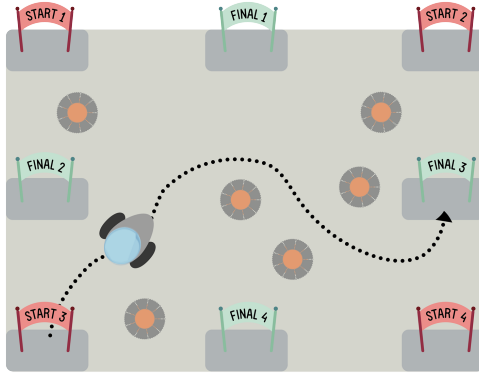
SSH, escritorio remoto, etc) para iniciar una prueba. No se permitirá tener dispositivos externos conectados al robot mientras se ejecuta la prueba.

Cuando se encienda un robot, éste deberá mostrar a través de su sistema de visualización que se encuentra listo para iniciar. Para saber si está listo, un robot debe solicitar un servicio al master llamado *ack_service* enviándole el número del grupo al que pertenece el robot (llamado *group_number* de tipo *(int32)*) y su dirección IP (llamado *ip_address* de tipo *(string)*). El *master* por su lado responderá a dicho servicio enviando un número llamado *state*, tipo *int32* con su estado. Si el número enviado por el *master* es cero (0), significa que aún no se encuentra listo para iniciar; por el contrario, si el número enviado es uno (1), significa que ya se encuentra listo y en ese momento el robot lo debe mostrar.

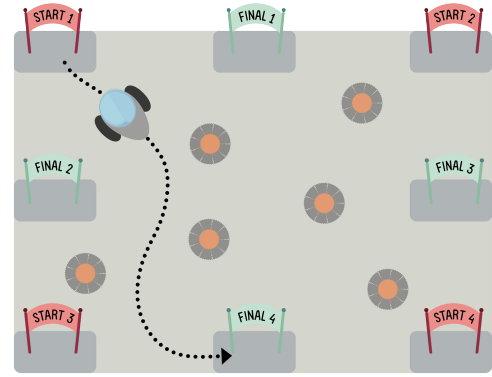
Una vez el robot se encuentre listo para iniciar, se quedará esperando que le den la orden para iniciar la prueba. Para esto, el *master* solicitará un servicio al robot llamado *start_service*. Cuando el robot reciba la solicitud del servicio el robot podrá iniciar la prueba. Al enviar la solicitud, el *master* enviará cuatro (4) argumentos: el primero se llamará *start*, será tipo *Pose* (del paquete de ROS *geometry_msgs*) e indicará la pose inicial del robot en el marco global de referencia, el segundo se llamará *goal*, será tipo *Pose* e indicará la pose final del robot en el marco global de referencia, el tercero se llamará *n_obstacles*, será tipo *int32* e indicará el número de obstáculos en la pista y el último se llamará *obstacles_array*, será de tipo *Obstacle[]* y será un arreglo de tamaño *n_obstacles* donde en cada posición habrá un elemento por cada obstáculo de la pista. El tipo *Obstacle* estará definido dentro del paquete *master_package* y tendrá dos campos: el primero se llamará *position*, será tipo *Pose* y tendrá la posición del obstáculo en el marco global de referencia, el segundo se llamará *diameter*, será un *int32* y tendrá el diámetro del obstáculo en milímetros.

Mientras el robot se mueve a través de la pista buscando su objetivo, deberá publicar periódicamente su pose. La frecuencia de publicación la definirá cada equipo de manera que el *master* pueda conocer la pose - posición y orientación - del robot en tiempo real. El robot publicará esta información en un tópico llamado */robot_position* en un mensaje tipo *Pose*. También deberá publicar de manera periódica la estimación de la incertidumbre relacionada con su movimiento en un tópico llamado */robot_uncertainty*. Este tópico recibirá mensajes tipo *Covariance* los cuales estarán definidos en el paquete *master_package* y tendrá nueve (9) campos tipo *float32* llamados σ_{ij} donde i y j toman valores del uno (1) al (3) y corresponden a $i = 1$ para la posición x , $i = 2$ a la posición y e $i = 3$ a θ . Al lograr su objetivo, el robot enviará un servicio al *master* llamado *end_service* con la información requerida¹. Al recibirlo, el

¹Ver la Sección 3.4 para más información sobre el tipo de mensaje que deberá enviar el robot



(a) Ejemplo 1 de configuración



(b) Ejemplo 2 de configuración

Figura 2: Posibles representaciones del escenario donde se llevarán a cabo las pruebas

master finalizará la prueba. En caso de que el robot no logre finalizar el objetivo de la prueba antes de que termine el tiempo máximo, el *master* finalizará la prueba sin recibir el servicio.

3.3. El Escenario

El escenario en el que se llevarán a cabo las pruebas, consta de un rectángulo en el que estarán demarcados varios puntos de inicio y de fin, así como de obstáculos cilíndricos. El tamaño exacto del escenario aún está por definir y será comunicado a través de Sicua+. La posición exacta de los obstáculos y de los puntos iniciales y finales no serán conocidas por los equipos sino hasta el momento de iniciar la prueba. En el momento en el que se publiquen las dimensiones exactas, también se mostrará el sistema coordenado del marco de referencia global y los posibles puntos iniciales y finales. Junto con esta información también se publicará el material exacto del escenario y éste estará disponible antes del día de la prueba final.

La Figura 2a y la Figura 2b muestran dos posibles representaciones del escenario en el que se moverán los robots. **Las representaciones mostradas en las figuras sobre el escenario son únicamente para mostrar posibles configuraciones de posiciones iniciales y finales del robot, así como de la posición de los obstáculos. Éstas están sujetas a cambios y su versión definitiva será publicada en Sicua+.**

3.4. El Reto

El robot móvil construido por cada equipo iniciará en un punto dado del escenario y deberá trasladarse desde allí hasta el punto final designado sin golpear ningún obstáculo en su trayectoria y localizándose durante todo el recorrido (ver Sección 3.2 para más información sobre la forma en la que se le comunicará al robot la información sobre el escenario). En el punto final se encontrará una contraseña de cuatro (4) dígitos que el robot deberá identificar usando una cámara web y enviar al *master* a través del servicio *end_service*. Este servicio tendrá un campo tipo *int32* llamado *password* que contendrá el número de cuatro (4) dígitos encontrado por el robot. El *master* contestará dicho servicio usando un *int32* llamado *correct* que tendrá el valor de uno (1) si la contraseña enviada fue correcta y cero (0) de lo contrario. En caso de error, el robot tendrá la oportunidad de intentarlo de nuevo mientras el tiempo no haya

terminado.

La contraseña que deberá recolectar el robot será un número de cuatro (4) dígitos escrito a mano y se encontrará a la vista (sin oclusiones) del robot si éste se encuentra posicionado correctamente en el punto final de llegada. En cualquier caso, el objetivo principal de la prueba consiste en ***llegar de la posición inicial a la posición final con un error de pose bajo, sin golpear obstáculos, manteniendo un seguimiento de su localización, sin usar sensores distintos a los que tiene a bordo el robot, y encontrar y transmitir la contraseña secreta al master, en el menor tiempo posible.***

Para solucionar el reto planteado, la solución de los equipos deberá contener al menos los siguientes elementos:

- **Robot diferencial:** Cada equipo deberá diseñar y construir un robot diferencial que cumpla con las restricciones indicadas en la Sección 3.1
- **Control de las ruedas:** Cada equipo deberá diseñar e implementar un sistema que le permita controlar la velocidad de cada una de las ruedas del robot. Las especificaciones y el tipo de control deberán ser dados por cada equipo
- **Cinemática del robot:** Cada equipo deberá implementar la cinemática del robot construido
- **Ley de control:** Cada equipo deberá implementar una ley de control de lazo cerrado que le permita llevar al robot desde una pose inicial hasta una pose final en el marco de referencia global.
- **Planeación de rutas:** Cada equipo deberá implementar un algoritmo de planeación de rutas que le permita al robot encontrar una ruta libre de obstáculos desde un punto inicial hasta un punto final en el escenario
- **Localización:** Cada equipo deberá implementar la teoría vista de Localización usando Filtro de Kalman para obtener una estimación de la posición del robot en cada movimiento, así como también la estimación de la incertidumbre acumulada
- **Reconocimiento de dígitos:** Cada equipo deberá implementar un programa que use la información obtenida por la cámara a bordo del robot y reconozca la contraseña de dígitos escritos a mano ubicada en la posición final de la prueba

La correcta y apropiada integración de los elementos descritos anteriormente permitirá a los equipos la solución del reto presentado.

4. Competencia

El día de la sustentación del proyecto² cada equipo tendrá dos oportunidades de realizar la prueba y en cada una de estas, los jurados tomarán el puntaje basado en el desempeño del robot. El puntaje de la prueba se calculará como una combinación de los siguientes parámetros:

- Si la prueba fue finalizada adecuadamente o no³

²Aún está pendiente definir la fecha exacta y dependerá en gran medida de la planeación de exámenes finales dada por Admisiones y Registro

³Se declara finalizada correctamente la prueba cuando el robot envía la contraseña correcta al *master*, éste la verifica y responde dando por finalizado el contador de tiempo

- Tiempo en completar adecuadamente la prueba
- Error de posición y orientación
- Número de obstáculos chocados durante la ejecución de la prueba

Además del puntaje obtenido, un robot puede ser penalizado por aspectos tales como hacer daños a la pista o no seguir las indicaciones del *master* de manera correcta. Estas penalizaciones las decidirán los jueces.

Buena Suerte!