

# Trabajo Práctico Especial - Robótica Móvil

## Robótica Móvil - Un enfoque probabilístico

---

Prof. Dr. Ignacio Mas

10 de agosto de 2020

**Fecha límite de entrega:** 7/09/20.

**Modo de entrega:** Reporte escrito (via Campus), presentación oral (via Google Meet) y demostración en un robot real.

### 1. Introducción

El objetivo de este trabajo es demostrar el manejo de conocimientos en el área de la robótica móvil que permitan utilizar eficientemente herramientas para la resolución de problemas reales. Se espera que se trabaje de manera libre, esto es, eligiendo de manera criteriosa los métodos y las herramientas necesarias para resolver las distintas problemáticas y procurando los datos e información no provista.

Dadas las restricciones de movilidad actuales, los algoritmos deberán verificarse durante su desarrollo en un simulador puesto a disposición por la cátedra, que puede ser modificado y/o mejorado, siempre y cuando mantenga su compatibilidad con la plataforma robótica real con la cual los resultados serán finalmente evaluados.

La participación es **optativa**. Participar y alcanzar con éxito los objetivos que se proponen permitirá la evaluación de la asimilación de los contenidos de la materia y –en función de lo que determine la Facultad de Ingeniería de la UBA como métodos válidos de aprobación de materias– simplificar la evaluación final de cada participante.

Quienes elijan no participar de este proyecto deberán rendir un examen final según las pautas que se determinen oportunamente por la cátedra en función de lo que dictamine la Institución.

### 1.1. Condiciones para participar

Podrán participar grupos de 1, 2 o 3 alumn@s, quienes deben tener la cursada aprobada. Cada grupo que desee participar deberá declararse como tal inscribiéndose en el formulario '*Inscripción para participar en TP Especial*' del Campus de la materia e ingresando allí el nombre de sus integrantes y el nombre del grupo. Este paso debe completarse antes del 13 de agosto de 2020.

### 1.2. Condiciones para aprobar

Habrán 3 etapas que deberán ser completadas:

- **Reporte** Deberá presentarse un reporte escrito de los desarrollos realizados. Pueden ser suplementados con videos realizados con el simulador. Se entregará via Campus.
- **Presentación** Se realizará una presentación por videoconferencia de entre 10 y 15 minutos –al que asistirán representantes de la cátedra y los demás grupos– detallando el sistema propuesto.
- **Competencia** Se entregará el código desarrollado a la cátedra para ser ejecutado en un robot real y así evaluar la performance del sistema. Este evento será transmitido por videoconferencia.

### 1.3. ¿Qué beneficios obtengo por participar?

Ser parte de un evento histórico. Y cumplir los requerimientos para la aprobación de la materia (sujeta a las disposiciones institucionales oportunamente dictadas).

### 1.4. ¿Qué pasa si nada funciona al momento de la Competencia?

Dado que esto es una iniciativa piloto que esta lejos de haber sido verificada para certificar su correcto funcionamiento, existe la posibilidad de que la etapa experimental no cumpla con los objetivos esperados. Es de vital importancia asumir una actitud positiva de cara a como pueda desarrollarse este proyecto y contemplar la posibilidad de que los eventos no fluyan de la forma más suave. Aquellos que manifiesten su frustración en el caso de no lograr la funcionalidad esperada en el robot real, seguramente tendrán la posibilidad de mostrar sus logros en ediciones futuras de este evento a realizarse de forma presencial en la facultad durante la nueva normalidad post-pandemia. En cuanto a la evaluación, si dificultades técnicas coartasen la posibilidad de demostrar lo desarrollado en el robot real, se ponderará lo realizado en el reporte y la presentación para calificar el desempeño de los participantes.

### 1.5. ¿Qué herramientas tengo para trabajar?

Los grupos tendrán acceso a las siguiente herramientas:

### 1.5.1. Simuladores

La cátedra pone a disposición de los grupos dos simuladores, uno en MATLAB y otro en Python. Las características y el modo de funcionamiento de cada uno pueden ser discimiles. Hay que arremangarse. Ambos tendrán una interfaz gráfica que facilitará su uso. El trabajo realizado deberá respetar ciertas estructuras que permitan una transición directa entre el simulador y la plataforma robótica real.

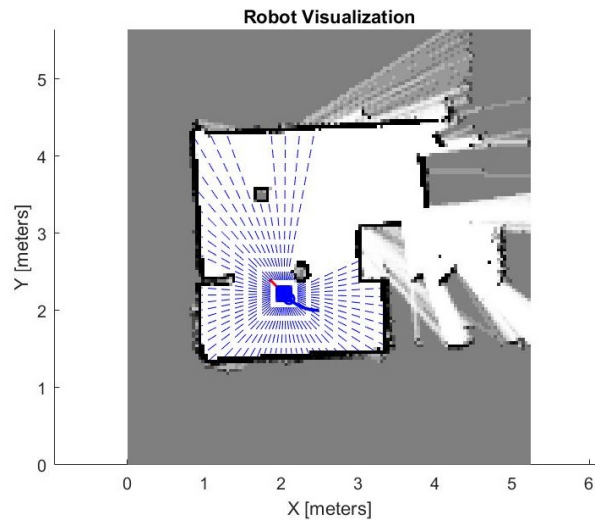


Figura 1.1: Interfaz gráfica del simulador en lenguaje MATLAB.

Una tercera opción como herramienta de desarrollo será el entorno ROS melodic (ros.org). Esta opción solo se recomienda para aquellos que tengan cierta experiencia en el desarrollo de software para robots y la cátedra no proveerá soporte alguno.

### 1.5.2. Robot móvil

Los algoritmos serán aplicados a un robot real, que está compuesto por una plataforma de aspiradora doméstica tipo Roomba de configuración diferencial, con encoders que estiman la pose y un sensor tipo lidar de 360 grados de cobertura. El robot aceptará comandos de velocidad lineal y angular dentro de un rango permitido y entregará lecturas de odometría utilizando la técnica de dead-reckoning y lecturas del sensor lidar con una resolución angular de  $2.5^\circ$ .

## 1.6. Especificaciones del robot

El robot esta basado en una plataforma comercial iRobot Roomba serie 600 de manejo diferencial. Cuenta con un lidar YDLIDAR X4 y encoders.

El lidar está ubicado a 18cm de altura (con respecto al piso) y 9cm por delante del centro del robot (que se define como el centro del círculo de 35cm de diámetro y coincide con el



Figura 1.2: Interfaz gráfica del simulador en lenguaje PYTHON.

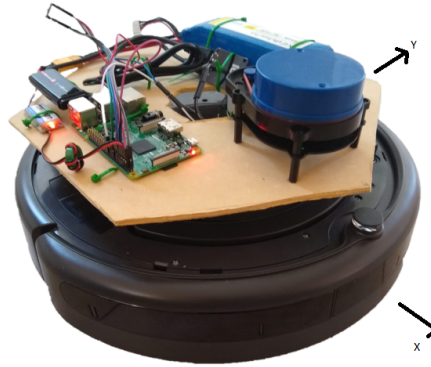


Figura 1.3: Robot diferencial con sensor LIDAR.

punto medio del eje que contiene ambas ruedas). El lidar esta orientado mirando ‘hacia atrás’, es decir, el ángulo cero corresponde a la medición que mira hacia el centro del robot y las mediciones progresan en sentido horario. Las mediciones inválidas toman el valor cero.

La velocidad lineal del robot es positiva hacia adelante y negativa hacia atrás. La velocidad angular es positiva en sentido anti-horario.

La siguiente tabla detalla las características del robot:

Característica	valor	unidad
Diámetro total	35,0	cm
Distancia entre ruedas	23,5	cm
Diámetro de las ruedas	7,2	cm
Comandos de velocidad	(v,w)	m/s
Rango permitido de velocidad lineal v	(-0,5...0,5)	m/s
Rango recomendado de velocidad lineal v	(-0,1...0,1)	m/s
Rango permitido de velocidad angular w	(-4,25...4,25)	rad/s
Rango recomendado de velocidad angular w	(-0,1...0,1)	rad/s
Lidar, rango max.	10	m (aprox.)
Lidar, rango min.	0.1	m (aprox.)
Lidar, cantidad de puntos	144	
Lidar, ubicacion en terna del robot	(0,09 0 0.18)	m
Lidar, orientación con respecto a terna del robot	$\pi$	rad
Tasa de muestreo (dt)	0,1	s

### 1.7. Detalles de los simuladores

Los simuladores intentan principalmente reflejar la interfaz con el robot real y deben ser sintonizados para obtener una funcionalidad similar a éste.

El simulador en MATLAB será subido al Campus y el simulador en Python puede ser obtenido utilizando git desde <https://gitlab.com/javierluiso/proboticssim>.

Los simuladores incluyen un mapa que será usado para una de las actividades (aunque puede llegar a ser reemplazado por otro con posterioridad).

Ambos simuladores están en período de desarrollo y pueden llegar a ser actualizados por la cátedra mientras se desarrolla este proyecto.

El simulador MATLAB funciona en R2020a y en R2016b. Se espera que no se utilicen versiones de MATLAB anteriores a R2016b. La compatibilidad con versiones distintas a las mencionadas deberá ser verificada.

El simulador Python y el desarrollo realizado deben ser compatibles con PYTHON 3.x para mantener compatibilidad con la interfaz del robot real.

## 2. Desafío 1: Patrullaje

En este escenario, el robot debe visitar 3 lugares en un entorno conocido en el orden propuesto. El robot comienza su tarea en algún lugar del entorno (no definido a priori) y debe visitar los tres puntos definidos por las siguientes coordenadas (en metros):

Punto A:	(2,9 ; 3,7)
Punto B:	(1,3 ; 3,7)
Punto C:	(1,5 ; 1,9)

El robot debe permanecer con velocidad cero por 3 segundos al llegar a cada punto intermedio y detenerse en el punto final.

El robot tiene 3 minutos como máximo para realizar la tarea y no puede chocar con ningún obstáculo mientras la realiza. Si el robot detecta una colisión, su sistema interno deshabilitará sus motores y la misión habrá fracasado.

La tarea será calificada según los puntos sean visitados o no y según el tiempo que conlleve finalizar la tarea.

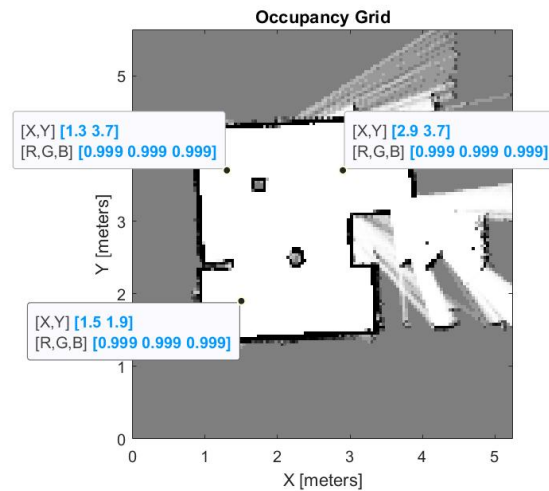


Figura 2.1: Coordenadas por las que debe pasar el robot.

### 3. Desafío 2: Exploración

En este escenario, el robot se despierta en un entorno desconocido y debe generar un mapa de ocupación del mismo, explorando el entorno. Si el robot detecta una colisión deshabilitará sus motores y la misión habrá fracasado.

El robot tiene 3 minutos como máximo para realizar la tarea. Al finalizar ese tiempo el mapa generado debe ser guardado para su posterior comparación con el mapa real. El grupo cuyo mapa se asemeje en mayor grado al real obtendrá la mayor cantidad de puntos.

El entorno será –en cuanto a tamaño, proporciones y características– similar al mapa del Punto 2, y aún lo suficientemente distinto como justificar la creación de un nuevo mapa.

### 4. Comentarios adicionales

Algunos comentarios que pueden ser de utilidad:

1. Límites de velocidad: Más allá de los límites del robot, se recomienda mantener las velocidades en valores bajos, para mantener la integridad física del sistema y para

- minimizar los errores de medición (debido a vibraciones, etc).
2. Choque y afuera: el robot posee un detector de contactos que desactivará los motores si detecta una colisión. Si eso sucede, la ejecución es detenida y el sistema debe ser reconfigurado.
  3. Duración de las actividades. Para mantener la brevedad y permitir la participación de todos los grupos, las actividades tienen un límite estricto de tiempo. De todas formas, se espera que el tiempo acordado sea suficiente para completar la tarea sin inconvenientes. Si este no fuese el caso, se podrán modificar los criterios definidos.
  4. Cantidad de ejecuciones. Inicialmente se permitirá una ejecución por equipo. Llegado el caso, se podrán realizar más ejecuciones si el tiempo lo permite. Si el grupo desarrolla más de un método para cumplir una tarea, podrá ejecutar una en su turno, y las demás luego de que los otros grupos hayan participado.
  5. Se conformará una tabla de posiciones con los puntos obtenidos por cada grupo para fomentar la sana competencia.
  6. A través de la plataforma Campus se podrán realizar consultas y discusiones.