

Trabajo Práctico Especial - Robótica Móvil

Robótica Móvil - Un enfoque probabilístico

Prof. Dr. Ignacio Mas

13 de julio de 2022

Fecha límite de entrega: 4/08/22.

Modo de entrega: Reporte escrito y código (4/8/22, via Campus), presentación oral (semana del 8/8/22, a confirmar, Presencial @ FIUBA) y demostración en un robot real (semana del 15/8/22, a confirmar, @ FIUBA, donde se ejecutarán los códigos desarrollados).

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.

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 facilitar 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 cátedra¹.

¹Quien no participe que avise con tiempo porque la metodología alternativa de final requiere una inversión igual o mayor de tiempo que este proyecto.

1.1. Condiciones para participar

Podrán participar grupos de hasta 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 11 de julio de 2022.

1.2. Condiciones para aprobar

Habrán 3 etapas que deberán ser completadas:

- **Reporte** Deberá presentarse un reporte escrito de los desarrollos realizados y el código final. Pueden ser suplementados con videos realizados con el simulador. Se entregará via Campus.
- **Presentación** Se realizará una presentación presencial de entre 10 y 15 minutos –al que asistirán representantes de la cátedra y los demás grupos– detallando el sistema propuesto.
- **Competencia** Se ejecutará el código desarrollado en un robot real en un evento presencial en FIUBA y se evaluará la performance del sistema en cuanto a funcionalidad y en comparación de capacidades en relación a lo desarrollado por los otros grupos.

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 esta etapa del proyecto y contemplar la posibilidad de que los eventos no fluyan de la forma más suave. Esta será la primera edición de este trabajo final en que se trate de ejecutar los códigos de forma presencial en FIUBA. Se realizarán eventos de práctica y calibración previo a la competencia para quienes tengan un sistema totalmente funcional sobre el simulador. En cuanto a la evaluación, si dificultades técnicas coartasen la posibilidad de demostrar lo desarrollado en el robot real, se ponderará lo realizado con el simulador y descrito 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. Simulador

La cátedra pone a disposición de los grupos un simulador en MATLAB, que tendrá 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.

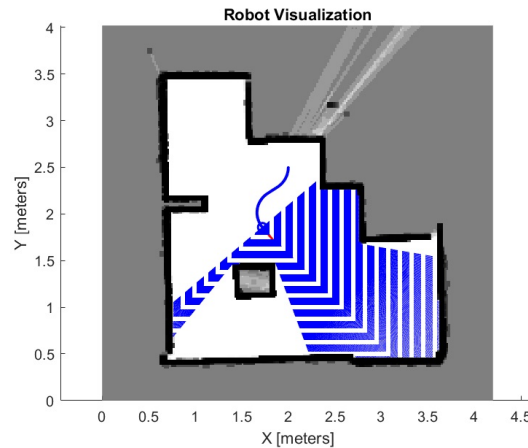


Figura 1.1: Interfaz gráfica del simulador en lenguaje MATLAB.

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 180 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 0.3516° .

1.6. Especificaciones del robot

El robot esta basado en la plataforma de desarrollo Kobuki de manejo diferencial. Cuenta con un lidar Hokuyo URG-04LX-UG01 y encoders.

El lidar está ubicado a aproximadamente 15cm de altura (con respecto al piso) y sobre el centro del robot (que se define como el centro del círculo de 35cm de diámetro y coincide con el punto medio del eje que contiene ambas ruedas). El lidar esta orientado mirando 'hacia adelante'. Las mediciones progresan en sentido horario comenzando en $-\pi$ y finalizando en π , entregando 513 puntos por medición. El código de MATLAB (tanto en el simulador como en el robot real) toma una de cada 3 mediciones para alivianar la carga computacional, resultando en 171 mediciones, pero esto puede ajustarse si se lo desea) Las mediciones inválidas toman el valor NaN.

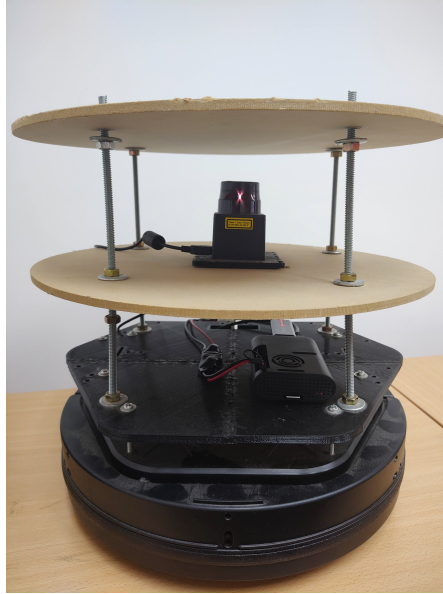


Figura 1.2: Robot diferencial con sensor LIDAR.

El robot tiene dos parantes que resultan en mediciones de alrededor de 15 o 17 cm, que deben ser consideradas ruido y ser filtradas (puede considerarse que cualquier medición menor a 20cm es errónea).

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	(-0,15... 0,15)	m/s
Rango permitido de velocidad angular w	(-4,25... 4,25)	rad/s
Rango recomendado de velocidad angular w	(-0,5... 0,5)	rad/s
Lidar, rango max.	5	m (aprox.)
Lidar, rango min.	0.06	m (aprox.)
Lidar, cantidad de puntos	171 (513/3)	
Lidar, ubicacion en terna del robot	(0,0 0,0 0,15)	m
Lidar, orientación con respecto a terna del robot	0	rad
Tasa de muestreo (dt)	0,1	s

1.7. Detalles del simulador

El simulador intenta principalmente reflejar la interfaz con el robot real y puede necesitar ser sintonizado para obtener una funcionalidad similar a éste.

El simulador en MATLAB será subido al Campus.

El simulador incluye un mapa que será usado para una de las actividades (aunque puede llegar a ser reemplazado por otro con posterioridad). El simulador esta en período de desarrollo y puede llegar a ser actualizado 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.

2. Desafío 1: Vigilancia

En este escenario, el robot debe vigilar una parte de la Facultad de Ingeniería y para esto debe visitar 2 lugares en un entorno conocido en un orden propuesto. El robot comienza su tarea en algún lugar del entorno (no definido ni conocido a priori) y debe visitar los dos puntos definidos por las siguientes coordenadas (en metros):

Punto A:	(1.5m ; 1.3m)
Punto B:	(4.3m ; 2.1m)

El robot debe permanecer con velocidad cero por 3 segundos al llegar al primer punto 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.

ATENCIÓN: El enunciado ha sido actualizado el 13/7 con un nuevo mapa de una sección de laboratorio LAR, en el 1er piso de la facultad, lugar donde finalmente se probarán los algoritmos. Los desarrollos deben focalizarse en este nuevo mapa pero a la vez deben ser flexibles para acomodar cualquier posible actualización futura.

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. El robot no debe colisionar con nada en su entorno (de hecho, el robot real detecta colisiones y se deshabilitan sus motores cuando esto sucede, haciendo fracasar la misión).

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.

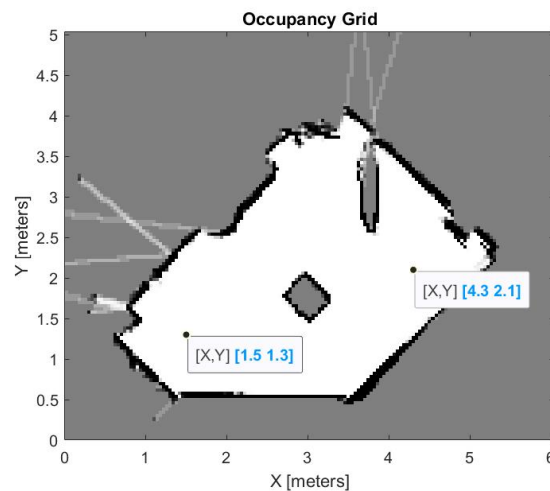


Figura 2.1: Coordenadas por las que debe pasar el robot.

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 para 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. Considerar la complejidad computacional como una de las variables de diseño, ya que es esperable que el sistema funcione en tiempo real (o cerca).
4. 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.

5. Cantidad de ejecuciones. Inicialmente se permitirá una ejecución por equipo durante la competencia. 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.
6. Se conformará una tabla de posiciones con los puntos obtenidos por cada grupo para fomentar la sana competencia.
7. A través de la plataforma Campus se podrán realizar consultas y discusiones.
8. Se definirán una o dos fechas de actividad presencial para ajustar los desarrollos realizados en el simulador a la plataforma robótica real. Se espera que este paso suceda cuando los algoritmos ya están corriendo de forma exitosa en el simulador.