

Trabajo Práctico Especial - Robótica Móvil

Parte A

Robótica Móvil

Prof. Dr. Ignacio Mas

22 de noviembre de 2024

Fecha límite de entrega: 1/12/24, 23:59hs.

Modo de entrega: Enviar por el Aula Virtual del Campus **en un solo archivo comprimido** el código comentado y traer el código al laboratorio el 2/12/24 a las 10am, y demostrar en un robot real que el algoritmo funciona.

Modalidad de trabajo: Trabajo Individual.

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 averiguando 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 estimar 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.

1.1. Condiciones para participar

Los alumnos que participen deben tener la cursada aprobada, o sea, todos los TPs de cursada completos de forma satisfactoria.

1.2. Condiciones para aprobar

- **Código** Deberá enviarse a través de Campus Virtual un archivo comprimido con el código desarrollado. Puede ser suplementado con videos realizados con el simulador.
- **Presentación:** Se ejecutará el código desarrollado en un robot real en un evento presencial en FIUBA el **lunes 2 de diciembre a las 10am** y se evaluará la performance del sistema en cuanto a funcionalidad y en comparación de capacidades en relación a lo desarrollado por los otros alumnos.

1.3. ¿Qué beneficios obtengo por participar?

Completar esta Parte A del Trabajo Final permite participar de la Parte B, que será el verdadero desafío que permitirá la evaluación final del curso.

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

Dado que esto es una iniciativa que implica lidiar con situaciones inciertas (por ejemplo, que el robot no funcione) 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 que uno desea.

1.5. ¿Qué herramientas tengo para trabajar?

Los grupos tendrán acceso a las siguientes herramientas:

1.5.1. Simulador

La cátedra pone a disposición 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.

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

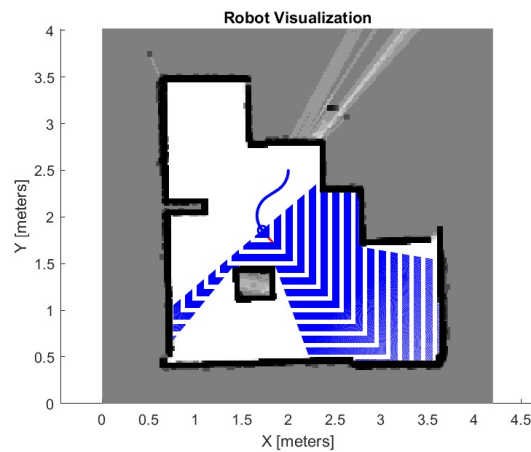


Figura 1.1: Interfaz gráfica del simulador en lenguaje MATLAB.

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 accionamiento diferencial, con encoders que estiman la odometría 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° .



Figura 1.2: Robot diferencial Kobuki con sensor LIDAR Hokuyo y computadora embebida Raspberry Pi.

1.6. Especificaciones del robot

El robot está 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 20cm de altura (con respecto al piso), en las coordenadas $x = 7cm$, $y = 0cm$ de las ternas del robot (ver figura 1.3), cuyo origen se encuentra en el centro del círculo de 35cm de diámetro y coincide con el punto medio del eje que contiene ambas ruedas). El lidar está orientado mirando 'hacia adelante'. Las mediciones progresan en sentido horario comenzando en $-\pi/2$ y finalizando en $\pi/2$, 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.

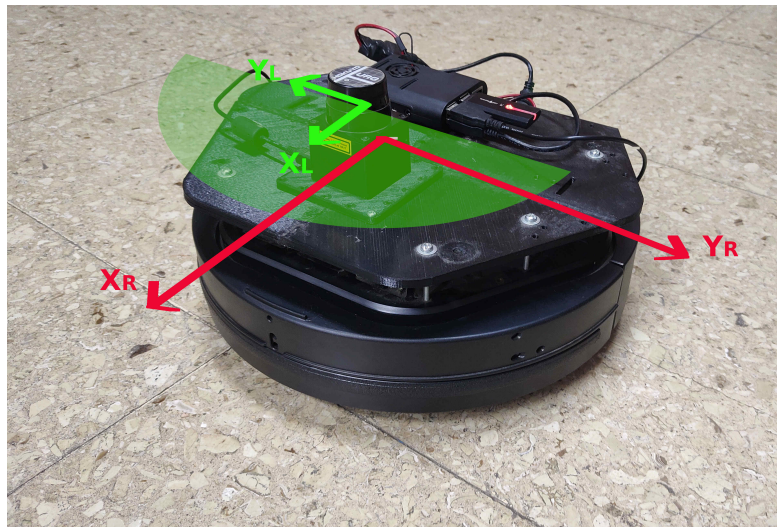


Figura 1.3: Robot con coordenadas locales del robot y campo de visión y coordenadas del LI-DAR.

La velocidad lineal del robot es positiva hacia adelante y negativa hacia atrás. La velocidad angular es positiva en sentido anti-horario. La lectura de odometría es la integración simple de la velocidad del robot, es decir, el robot entrega una lectura de odometría que es la integración de velocidades desde que se encendió el robot. Por lo tanto, la manera correcta de tomar una medición de odometría es calcular la diferencia entre la posición actual y la del tiempo anterior dada una iteración del algoritmo.

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	(0,07 0,0 0,0)	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 este.

El simulador está desarrollado en MATLAB y será subido al Campus para su uso.

El simulador incluye un mapa que será usado para una de las actividades (y que potencialmente puede ser reemplazado por otro de mayor calidad). 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 fue desarrollado en R2020a y verificado en versiones R2016b o más nuevas. Se espera que no se utilicen versiones de MATLAB anteriores a R2016b. La compatibilidad con versiones distintas a las mencionadas deberá ser verificada.

2. El Desafío: Navegación libre de choques

Esta primera parte del Trabajo es simple y permitirá familiarizarse con la plataforma de desarrollo para poder encarar la Parte B del trabajo de forma más directa.

El objetivo es que el robot pueda moverse por un entorno desconocido sin chocar con obstáculos. Se espera que se mueva a una velocidad que puede ser constante o variable (se recomienda evitar cambios bruscos de velocidad) por el espacio no ocupado del entorno y evada tanto paredes como obstáculos dinámicos que encuentre en su camino.

Esto resultará en un comportamiento de exploración que será utilizado en la Parte B del trabajo para realizar tareas de localización y navegación. En esta parte del trabajo no es necesario construir un mapa o localizarse en el mapa del entorno. El comportamiento puede ser puramente reactivo.

El desarrollo se realizará con el simulador, utilizando como entorno cualquiera de los mapas incluidos en el mismo. La demostración se realizará con el robot en el ámbito de la facultad en un entorno de características similares a dichos mapas.

3. 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 deseable que el sistema funcione en tiempo real (o cerca).
4. Duración de las actividades. Para mantener la brevedad y permitir la participación de todos, 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 participantes desarrollan más de un método para cumplir una tarea, podrá ejecutar una en su turno, y las demás luego de que los demás hayan participado.
6. A través de la plataforma Campus se podrán realizar consultas y discusiones.