

Trabajo Práctico Final: Traxbot (Localización y Movimiento)

Inteligencia Artificial para Robots - CEIA

5 de octubre de 2021

1. Descripción del robot

Una compañía de robots llamada Trax ha creado una línea de pequeños robots autónomos, diseñados para atravesar el desierto en búsqueda de depósitos de agua.

Un Traxbot luce como un pequeño tanque. Cada uno mide alrededor de medio metro de largo y se traslada sobre dos orugas metálicas. Para manejarse, el Traxbot puede avanzar en línea recta o girar. Para hacer un giro a la derecha, el Traxbot avanza, para, gira 90° y luego continúa avanzando recto.

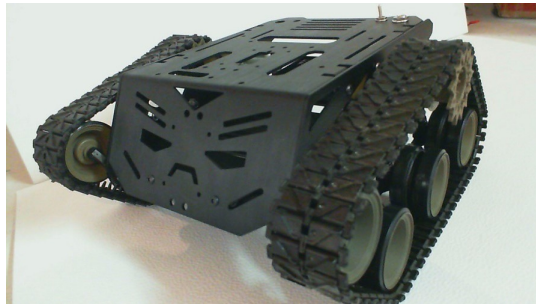


Figura 1: Algo similar al Traxbot.

2. Problema a resolver

En esta serie de problemas se deberá rescatar a un Traxbot perdido en algún lugar del desierto. Actualmente se encuentra girando prácticamente en círculo, avanza una cantidad de pasos, para, gira una cierta cantidad y repite el proceso. Por suerte, el Traxbot sigue enviando todos los datos de los sensores al cuartel general.

En este proyecto se comenzará con una simple versión del problema y se irá agregando complejidad gradualmente. Para el final, se tendrá un plan completamente articulado para recuperar el Traxbot perdido.

2.1. Primera parte - Localización

Vamos a comenzar considerando que el robot está girando de manera circular (o casi). Asumir que el robot vive en un plano de coordenadas (x, y) y, por ahora, está enviando mediciones del sensor con precisión absoluta (sin error).

Con algunas mediciones se debería poder determinar el paso y el ángulo con el que está girando el robot. Con esas dos piezas de código, se debería poder escribir una función que prediga la próxima ubicación del Traxbot.

Se puede utilizar la clase robot de *robot.py*.

2.1.1. Tu trabajo

Completar la función *estimate_next_pos*.

2.2. Segunda Parte - Localización

Ahora el escenario se vuelve más realista. Las mediciones del sensor del robot van a ser ruidosas (aunque aún el movimiento se mantiene libre de ruido). El objetivo de esta parte es escribir una función que tome como entrada la próxima medición ruidosa del sensor y su salida sea la mejor aproximación de la próxima posición del robot.

Tener en cuenta que el movimiento del robot en este caso es no lineal y en un ambiente continuo. ¿Qué filtros de los que vimos podrían implementarse?

2.2.1. Tu trabajo

Completar la función *estimate_next_pos*. Se considerará correcto si la estimación está dentro de los 0.01 *stepsizes* de la próxima posición del Traxbot.

2.3. Tercera Parte - Movimiento velocidad variable

Ahora el objetivo es recuperar el Traxbot. En esta parte del proyecto, la velocidad de tu robot será del doble del robot a atrapar, lo que significa que el parámetro de distancia de tu robot será del doble que el robot a atrapar. Tu robot puede moverse menos que este parámetro si así lo deseas para reducir la velocidad la velocidad de tu robot cuando esté cerca de atrapar al otro.

2.3.1. Tu trabajo

Completar la función *next_move*. Esta función te dará acceso a la posición y orientación de tu robot (el cazador); la medición más reciente recibida desde el robot a atrapar (el objetivo), la distancia máxima que tu robot se puede medir en un dado tiempo, y otra variable, llamada OTHER, que puedes utilizar para guardar información.

Tu función devolverá cuánto quieres que tu robot gire, la distancia que quieres que se mueva, y la variable OTHER, con cualquier información que quieras guardar.

2.4. Cuarta Parte - Movimiento velocidad constante

Nuevamente, tu objetivo es seguir el robot hasta atraparlo. Pero esta vez tu velocidad será la misma que el robot que sigues. Esto requerirá planear cuidadosamente cuales serán tus movimientos.

2.4.1. Tu trabajo

Completar la función *next_move*, de manera similar a como lo hiciste la última vez.