

Universidad de Alcalá

Grado en Ingeniería de Computadores

PERCEPCION Y CONTROL
Práctica de Laboratorio (PECL)
Curso 2024/2025



Sistema de Navegación y Mapeo para Robot Móvil

DNI – Apellidos, Nombre

52003666E – Valcarcel Alvarez, Adrian

12795971J – Pajuelo Condori, Joseph

Contenido

52003666E – Valcarcel Alvarez, Adrian 12795971J – Pajuelo Condori, Joseph	1
Introducción	3
Parámetros Principales	3
Funciones Principales	3
1. actualizar_mapa_completo	3
2. world2map	4
3. visualizar_mapa	4
4. guardar_mapa_final	4
Funciones de Navegación	5
1. avanzar_coordenadas	5
2. girar_coordenadas	5
3. girar_180	5
4. girar	5
Funciones de Soporte	6
1. min_distancia_lidar	6
2. distancia_lidar	6
3. avanzar	6
4. detener	6
5. obtener_posicion	6
6. obtener_yaw	6
Algoritmo de Exploración	7
Fase 1: Exploración Aleatoria	7
Fase 2: Exploración Dirigida	7
Conclusión	7

Introducción

Este documento describe en detalle el sistema de navegación y mapeo implementado en MATLAB para un robot móvil. El sistema permite al robot explorar un entorno desconocido, construir un mapa de ocupación basado en datos de odometría y LIDAR, y navegar hacia áreas inexploradas de manera autónoma.

Parámetros Principales

resolution: 0.06 (metros por píxel del mapa)

mapSize: 15/resolution (tamaño del mapa en píxeles)

logOddsMap: Matriz que almacena el mapa en formato log-odds

visitedMap: Matriz booleana que indica celdas visitadas

l_occ, l_free, l_min, l_max: Parámetros para actualización del mapa log-odds

distancia_umbral: 0.5m (distancia mínima para evitar obstáculos)

tiempo_limite: 120s (duración máxima de cada fase)

Funciones Principales

1. actualizar_mapa_completo

Propósito: Actualiza el mapa de ocupación basado en datos LIDAR y odometría.

Proceso:

Para cada medición del LIDAR (dentro del rango válido 0.15-8.0m):

Calcula la posición del punto detectado en coordenadas mundiales.

Traza una línea desde el robot hasta el punto detectado.

Actualiza las celdas de la línea como libres (l_free).

Actualiza la celda del punto final como ocupada (l_occ).

Mantiene los valores dentro de los límites [l_min, l_max].

Marca las celdas actualizadas como visitadas.

Parámetros:

x0, y0, theta: Posición y orientación actual del robot.

ranges, angleMin, angleInc: Datos del escaneo LIDAR.

resolution, mapSize: Parámetros del mapa.

logOddsMap, visitedMap: Mapas a actualizar.

l_occ, l_free, l_min, l_max: Parámetros de actualización.

2. world2map

Propósito: Convierte coordenadas mundiales (metros) a coordenadas de mapa (píxeles).

Proceso:

Escala las coordenadas por la resolución.

Ajusta el eje Y para que coincida con la representación matricial.

3. visualizar_mapa

Propósito: Muestra gráficamente el mapa de ocupación y la posición del robot.

Características:

Convierte log-odds a probabilidades para visualización.

Muestra la posición del robot como un punto azul.

Dibuja los rayos LIDAR como líneas rojas.

Usa una escala de grises (blanco=libre, negro=ocupado).

4. guardar_mapa_final

Propósito: Guarda el mapa final en un archivo .mat y lo visualiza.

Proceso:

Convierte log-odds a probabilidades.

Clasifica celdas como libres (<0.3), ocupadas (>0.7) o desconocidas.

Guarda el mapa en 'mapa_final_valores_0_05_1.mat'.

Muestra el mapa con leyenda clara.

Funciones de Navegación

1. avanzar_coordenadas

Propósito: Navega hacia coordenadas objetivo evitando obstáculos.

Algoritmo:

Calcula distancia y dirección al objetivo.

Ajusta velocidad proporcional a la distancia.

Si hay obstáculos cercanos, gira 180° y avanza temporalmente.

Usa control proporcional para mantener el rumbo.

Parámetros:

x_objetivo, y_objetivo: Coordenadas destino.

distancia_umbral: 0.5m (para detección de obstáculos).

tolerancia: 0.25m (para considerar destino alcanzado).

2. girar_coordenadas

Propósito: Orienta el robot hacia coordenadas objetivo.

Algoritmo:

Calcula ángulo hacia el objetivo.

Usa control proporcional para minimizar el error angular.

Se detiene cuando el error es menor a 0.05 rad ($\sim 3^\circ$).

3. girar_180

Propósito: Gira el robot 180° (anti-horario) a velocidad constante.

4. girar

Propósito: Gira el robot un ángulo aleatorio entre 0° - 90° .

Funciones de Soporte

1. min_distancia_lidar

Propósito: Devuelve la distancia mínima detectada por el LIDAR en un rango frontal (sensores 50-350 de 400 totales).

2. distancia_lidar

Propósito: Devuelve la distancia de un sensor LIDAR específico.

3. avanzar

Propósito: Mueve el robot hacia adelante a velocidad constante (0.3 m/s).

4. detener

Propósito: Detiene completamente el robot.

5. obtener_posicion

Propósito: Devuelve posición (x,y) y orientación (z) actual del robot.

6. obtener_yaw

Propósito: Extrae la orientación (yaw) del mensaje de odometría.

Algoritmo de Exploración

Fase 1: Exploración Aleatoria

Mientras no se alcance el tiempo límite:

Actualiza el mapa con datos LIDAR/odometría.

Si no hay obstáculos cercanos, avanza.

Si detecta obstáculo, gira 180°, avanza 1.5s, y gira ángulo aleatorio.

Visualiza el mapa en cada iteración.

Fase 2: Exploración Dirigida

Busca celdas inexploradas (valor 0.5) en el mapa.

Navega hacia la celda inexplorada más cercana.

Repite hasta explorar todo el mapa o alcanzar tiempo límite.

Conclusión

Este sistema implementa un enfoque completo para mapeo y navegación autónoma, combinando:

Mapeo probabilístico con log-odds

Evasión reactiva de obstáculos

Exploración aleatoria y dirigida

Visualización en tiempo real

El código es modular y cada función tiene un propósito específico, permitiendo fácil modificación y extensión para diferentes entornos o requisitos de navegación

