

Diseño y simulación de un robot autónomo con IA en Webots



Webots
robot simulation

Índice

1. DATOS DEL EQUIPO	3
2. OBJETIVO DEL PROYECTO	3
3. DISEÑO DEL SISTEMA ROBOTIZADO	3
4. CONTROL Y PROGRAMACIÓN	4
5. VISIÓN ARTIFICIAL E IA	5
6. PRUEBAS Y RESULTADOS.....	5
7. EVALUACIÓN DEL PROYECTO	5
8. EVIDENCIAS VISUALES	6
9. ARCHIVOS ADJUNTOS	6

1. DATOS DEL EQUIPO

- Nombre del proyecto: Pioner3-AT column mover
- Integrantes del equipo:
 1. Samuel Arteaga López
 2. Lucas Tarazona Comendador

2. OBJETIVO DEL PROYECTO

Describid brevemente cuál es la misión de vuestro robot. ¿Qué tarea tiene que cumplir? ¿Por qué es importante o interesante?

Nuestro robot tiene la misión de clasificar las columnas (obstáculos) del mapa según si es par o impar, basado en la suma del eje x e y del obstáculo.

3. DISEÑO DEL SISTEMA ROBOTIZADO

- **Tipo de robot elegido:** Móvil diferencial (Pioneer3-AT).
- **Razones de la elección:** Robot introductorio en visión artificial, robusto y sencillo de controlar, ideal para pruebas en entornos simulados.
- **Descripción del entorno simulado:** Entorno matricial sencillo creado con columnas en posiciones fijas, basado en el mapa por defecto de Webots.

- **Sensores y actuadores utilizados:**
- Cámara RGB
- Sensores de distancia (para evitar colisiones)
- Motores diferenciales (ruedas)

4. CONTROL Y PROGRAMACIÓN

- **Lenguaje de programación:** Python
- **Tipo de control usado:** Máquina de estados sencilla basada en detección visual.
- **Descripción del algoritmo principal:**
El robot recorre el entorno detectando columnas con la cámara. Al detectar una columna, calcula si la suma de su posición fila + columna es par o impar, y ejecuta la acción correspondiente (por ejemplo, moverse a izquierda si par, derecha si impar).

Mientras no termine el recorrido:

avanzar

si detecta columna:

capturar imagen

determinar posición (fila, columna)

calcular suma fila + columna

si suma es par:

realizar acción "par" (ej: mover columna a izquierda)

sino:

realizar acción "impar" (ej: mover columna a derecha)

evitar obstáculos cercanos

5. VISIÓN ARTIFICIAL E IA

- **Tipo de detección usado:** Detección por sensor (para reconocer las columnas)
- **Herramienta utilizada:** OpenCV
- **Comportamiento del robot ante lo detectado:**
 - Detecta los obstáculos por sensor (columnas).
 - Extrae la posición aproximada en el entorno.
 - Clasifica como par o impar y realiza movimiento de clasificación.

6. PRUEBAS Y RESULTADOS

- **Pruebas realizadas:**
 - Obstáculos estáticos (otras columnas, paredes)
 - **¿Qué funcionó bien?**
El avance del robot y más o menos la detección de las columnas
 - **¿Qué problemas tuvisteis?**
 - Detección errónea del sensor moviendo columnas más alejadas.
 - A veces interpretaba el suelo como objeto.
 - **¿Cómo los solucionasteis?**
Intentar ajustar el sensor de acercamiento

7. EVALUACIÓN DEL PROYECTO

- **Tiempo estimado de desarrollo:** 1 semana.
- **¿Se cumplió el objetivo?** Sí, el robot fue capaz de detectar columnas y clasificarlas correctamente en la mayoría de las pruebas.
- **¿Qué cambiaríais para mejorar el robot?**
 - Usar detección basada en forma además de color.
 - Implementar un algoritmo de localización más preciso para determinar la posición exacta.
- **¿Qué habéis aprendido del proyecto?**
 - Integrar visión artificial en un entorno simulado.

- Programar máquinas de estado básicas en Python.
- Planificar de manera estructurada un proyecto robótico completo.

8. EVIDENCIAS VISUALES

- **Capturas de pantalla del simulador:**
(Añadir capturas del robot moviéndose y detectando columnas)
- **Esquemas del entorno o diseño del robot:**
(Añadir un esquema sencillo del mapa, mostrando columnas y ruta del robot)
- **Vídeo o enlace:**
[Vídeo del proyecto](#) (enlace a vuestra grabación)

9. ARCHIVOS ADJUNTOS

- ☐ Código fuente
- ☐ Archivo del mundo Webots (.wbt)
- ☐ Controlador del robot (.py / .cpp)
- ☐ Presentación final (opcional)