

Descripción funcional y técnica del sistema

Visión general del comportamiento

El robot ejecuta un sistema autónomo basado en una **máquina de estados finitos** que le permite desplazarse entre habitaciones, identificar información visual semántica (números) situada en las paredes y atravesar puertas para continuar la exploración. El sistema integra **percepción geométrica con LIDAR, percepción visual basada en visión artificial e inteligencia artificial, y control cinemático reactivo**.

El ciclo general por habitación es:

1. Localización y centrado en la habitación (GOTO_ROOM_CENTER)
 2. Exploración visual de las paredes y lectura de números (TURN)
 3. Detección y aproximación a una puerta (GOTO_DOOR)
 4. Alineación precisa (ORIENT_TO_DOOR)
 5. Cruce de la puerta (CROSS_DOOR)
 6. Reinicio del ciclo en la siguiente habitación
-

Centrarse en la habitación (GOTO_ROOM_CENTER)

Al entrar en una habitación, el robot intenta estimar el **centro geométrico** a partir de las paredes detectadas por el sensor LIDAR. Para ello, se utiliza un módulo de percepción geométrica que analiza las líneas extraídas del entorno (`room_detector.estimate_center_from_walls`).

Gestión de incertidumbre del centro

- Si el centro **no puede estimarse de forma fiable**, el robot entra en un modo de exploración:
 - Gira sobre sí mismo con velocidad angular constante durante un intervalo de tiempo fijo.
 - Este comportamiento permite barrer la geometría de la habitación y estabilizar la detección de paredes.
- Este giro está temporalmente acotado para evitar oscilaciones indefinidas.

Aproximación al centro

Cuando el centro es detectado:

- Se calcula la distancia y el ángulo al centro en el sistema de referencia del robot.
- Se aplica un **control desacoplado**:
 - La velocidad angular es proporcional al error angular.

- La velocidad lineal se modula mediante un freno exponencial en función del ángulo, evitando trayectorias laterales inestables.
 - Al alcanzar una distancia inferior a un umbral (RELOCAL_CENTER_EPS), el robot se considera centrado.
 - En ese momento:
 - Se ajusta la pose del robot al centro nominal de la habitación.
 - Se reinicia la lógica de lectura de números.
 - Se transita al estado TURN.
-

Exploración visual y lectura de números (TURN)

En este estado, el robot gira sobre sí mismo durante un tiempo limitado para inspeccionar visualmente las paredes. Durante este giro se ejecuta de forma periódica un **pipeline de visión artificial y clasificación mediante red neuronal**.

El objetivo es detectar un **marco negro** en la pared y clasificar el dígito contenido en su interior.

Pipeline de visión artificial e inteligencia artificial

La percepción visual está diseñada como un proceso **jerárquico y eficiente**, combinando filtrado clásico por imagen con clasificación basada en aprendizaje profundo.

1. Adquisición de imagen

- Se obtiene una imagen RGB completa desde la cámara omnidireccional.
 - La imagen se convierte a escala de grises para reducir dimensionalidad y eliminar dependencia del color.
 - Se mantiene una copia para depuración visual.
-

2. Prefiltrado rápido (early exit)

Para evitar computación innecesaria:

- Se genera una versión reducida de la imagen.
- Se aplica un umbral de intensidad (`inRange`) para detectar regiones oscuras.
- Si no se detectan contornos negros significativos, se descarta el frame inmediatamente.

Este paso permite que el sistema sea **reactivo y eficiente en tiempo real**, incluso cuando no hay información relevante en la escena.

3. Detección del marco negro

Cuando el prefiltrado es positivo:

- Se aplica un suavizado gaussiano para reducir ruido.
- Se realiza una segmentación por intensidad para detectar regiones negras.
- Se aplican operaciones morfológicas de cierre para consolidar regiones fragmentadas.
- Se extraen contornos y se filtran por:
 - Área mínima relativa a la imagen
 - Relación de aspecto compatible con un marco aproximadamente cuadrado

Se selecciona el contorno más probable como **marco candidato**.

4. Extracción de la región interior (ROI)

- Se define una ROI interior al marco detectado, aplicando un margen para evitar bordes.
- Se descartan regiones con bajo contraste (desviación estándar baja), evitando clasificar ruido o sombras.

Esta etapa garantiza que solo se procesan regiones con **información visual suficiente**.

5. Segmentación del dígito

Dentro de la ROI:

- Se normaliza el contraste dinámicamente.
- Se aplica umbralización adaptativa gaussiana para separar el dígito del fondo.
- Se detectan contornos y se selecciona el contorno dominante:
 - Área significativa respecto a la ROI
 - No tocando los bordes (para evitar el marco)

El área seleccionada se considera el dígito.

6. Normalización para la red neuronal

- El dígito segmentado se reescala a **28×28 píxeles**.
- Se mantiene una única canalización en escala de grises.
- La imagen se normaliza a valores en el rango [0,1].

Esto permite una **compatibilidad directa con el dataset MNIST**.

7. Clasificación mediante red neuronal convolucional

El sistema utiliza una **red neuronal convolucional** entrenada previamente con MNIST:

- La red recibe un tensor de forma $(1, 1, 28, 28)$.
- Se ejecuta inferencia en CPU sin gradientes (`torch.no_grad()`).
- La salida se pasa por una función `softmax` para obtener probabilidades.
- Se selecciona el dígito con mayor probabilidad y su confianza asociada.

Solo se acepta la clasificación si la confianza supera un umbral (`CONF_THRESH`).

8. Gestión temporal de la información semántica

- El último dígito detectado se almacena junto con su confianza y timestamp.
- La interfaz Ice (`MNIST_getNumber`) devuelve el valor solo si:
 - El dígito es válido
 - No ha caducado (`MAX_AGE`)
- Esto evita reutilizar detecciones antiguas o falsas.

Además, el sistema garantiza que **solo se lee un número por habitación**.

Búsqueda y aproximación a puertas (GOTO_DOOR)

Tras finalizar TURN, el robot busca puertas mediante percepción LIDAR.

- Si no se detectan puertas, el robot gira para ampliar el campo perceptivo.
- Si hay puertas visibles:
 - Se selecciona la puerta más alineada con el eje frontal del robot.
 - Se calcula un punto previo a la puerta a una distancia fija.
 - Este punto se utiliza como objetivo intermedio para una aproximación estable.

El movimiento hacia este punto se realiza mediante un controlador cinemático suave.

Alineación y cruce de la puerta (ORIENT_TO_DOOR y CROSS_DOOR)

- En `ORIENT_TO_DOOR`, el robot ajusta su orientación hasta que el error angular con la puerta es inferior a un umbral.
- Una vez alineado, pasa a `CROSS_DOOR`, donde avanza de forma continua y sin reevaluación perceptiva.

- Tras cruzar:
 - Se actualiza la habitación actual.
 - Se reinician las variables internas.
 - Se vuelve a GOTO_ROOM_CENTER.