Introducción a la Robótica Móvil

Introducción a la Robótica Móvil: Índice

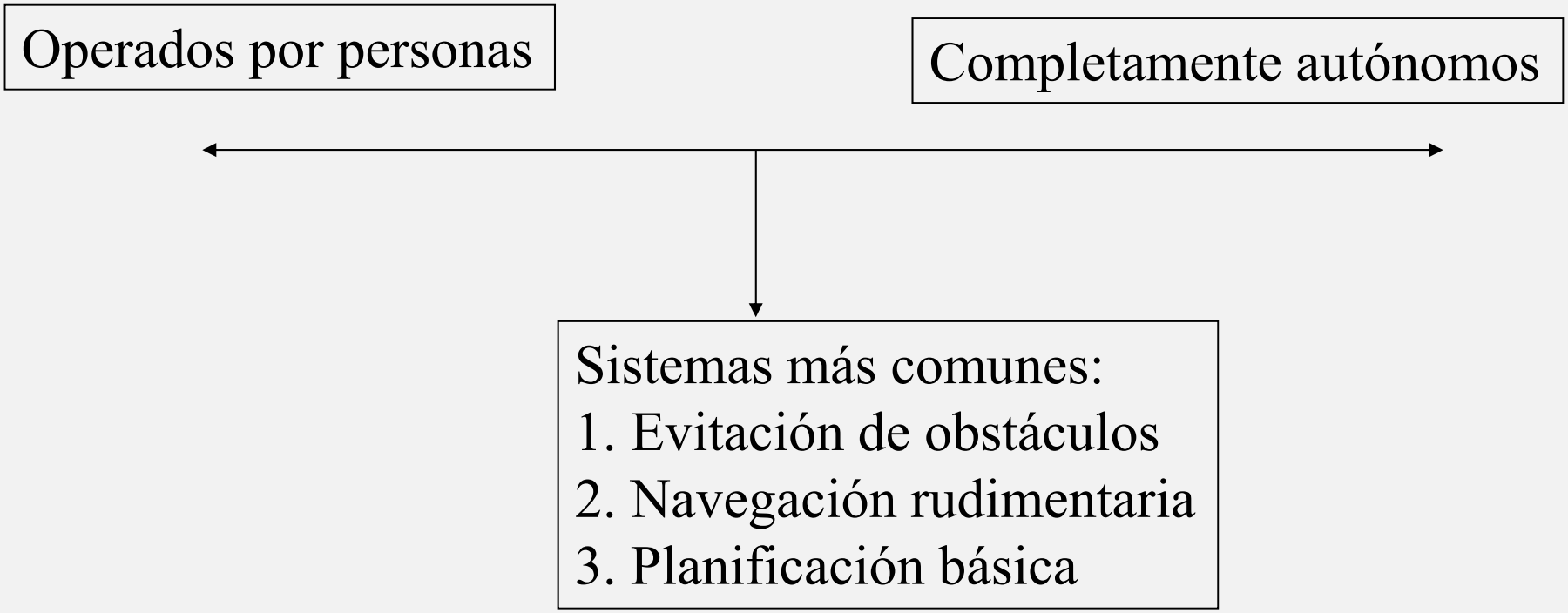
¿Qué es la Robótica Móvil ?

- El estudio de los robots que
  - se mueven respecto de su entorno
  - presentan *cierto grado* de autonomía cuando lo hacen

- ¿Cumple la definición de robot que propusimos?
  - un robot es una máquina **programable** con **partes mecánicas móviles controladas en lazo cerrado** (en tiempo real), que obtiene información del contexto mediante **sensores** y que actúa sobre el contexto mediante **actuadores**.
  - admite programar, en capas superiores de software, aplicaciones "inteligentes" que le permitan, por ejemplo:
    - tomar decisiones (en función del contexto)
    - aprender de su "experiencia"

¿Qué significa “cierto grado de autonomía"?

- La autonomía de los robots móviles es muy diversa:



Diversidad de los robots móviles

- Se pueden clasificar en diversas categorías:
  - modo de locomoción (piernas, ruedas...)
  - morfología (aspecto, tamaño...),
  - entorno de operación (interior/exterior, aéreo/terrestre/marino…)

¿Para qué sirven los robots móviles?

- Entes autónomos en movimiento:
  - coches, trenes, aviones, etc.
- Robocarros (AGV):
  - transporte y uso de herramientas y materiales en entorno industrial (pintura, soldadura...)
- Exploración
  - espacial, marina, minera, etc.
- Grandes industrias:
  - agricultura, fabricación, transporte, entretenimiento, etc.
- Actividades peligrosas
  - zonas contaminadas, desactivación de explosivos, etc.
- Soporte físico y/o cognitivo de personas con restricciones
  - Sillas inteligentes, robos de compañía, brazos articulados para manipulación

Subsistemas

- Hardware:
  - Mecánico
    - chasis y elementos móviles de locomoción (patas, ruedas, cadenas, etc.)
    - actuadores: propulsión, motores
  - Alimentación (generalmente) eléctrica
    - fuente de energía, control de potencia
  - Sensores
    - propioceptivos (encoders, acelerómetros, etc.)
    - exteroceptivos (sonar, radar, laser, etc.)
- Software
  - Control a bajo nivel
    - posición, velocidad, momento, etc.
  - Programación de tareas a alto nivel
    - Planificación, aprendizaje, etc.

Tareas a bajo nivel

- Control
  - Ajuste a condiciones cambiantes (p. e., cuando carga algo debe adaptarse al cambio de peso)
  - Problema dinámico: ¿qué momento hay que aplicar a cada rueda para comunicar al robot una determinada aceleración?
  - Control de bajo nivel de un robot con ruedas independientes: dirección y velocidad angular

- Cálculo de
  - movimientos suaves a velocidad y aceleración razonables
  - evitación de obstáculos y choques

Tareas a alto nivel

- Planificación de trayectorias
  - posición exacta del robot
  - trayectoria de una posición a otra expresada en función del tiempo
  - ¿Cuál es la mejor trayectoria según un determinado criterio (p. e., evitar obstáculos, mínima distancia, mínimo consumo…)?
- Corrección de trayectorias
  - Dado que el modelo del robot no es exacto, ¿qué podemos hacer para obtener buenos resultados, a pesar de los errores?
- Aprendizaje
  - Creación de mapas
  - Modificación del comportamiento, a partir de la experiencia

Arquitecturas de control

- *Top down*. Jerárquicas
  - Descomponer tareas en subtareas
  - Los niveles más altos trabajan en ciclos más largos
  - Ciclo: sensor-percepción-decisión-actuación

- *Bottom up*. Reactivas
  - Competencias a bajo nivel
  - Caminos directos entre sensores y actuadores

- Híbridas (arquitecturas de 3 niveles)
  - Algunas competencias en los niveles bajos
  - Planificación a nivel alto
  - Un nivel de comunicación para mediar entre ellos

Modelos matemáticos en robótica móvil

El control de los robots parte de modelos matemáticos bien definidos que

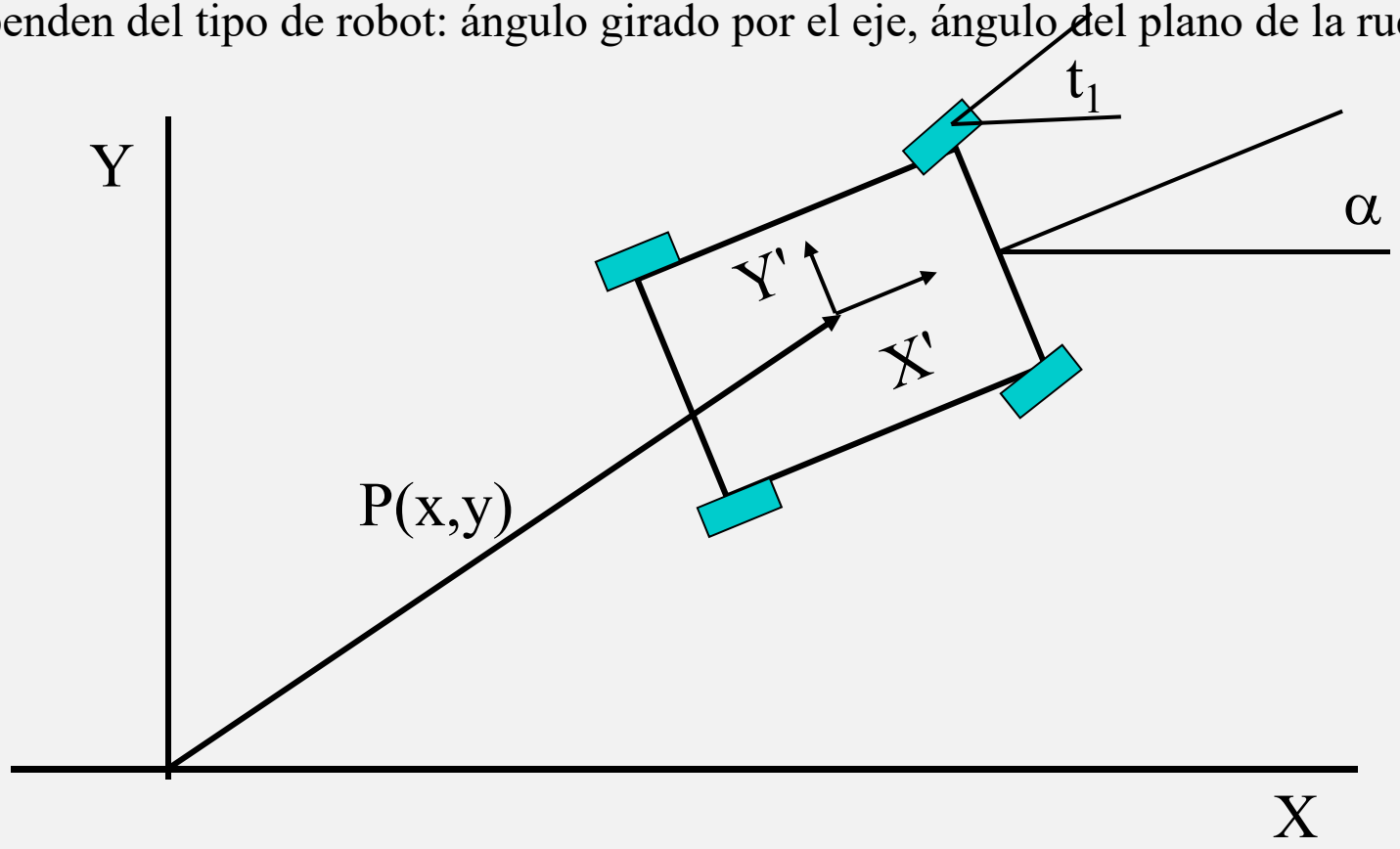
- parten de los parámetros articulares y espaciales del robot para calcular las trayectorias y su control
- tienen en cuenta las dimensiones y pesos de los elementos de cada robot y, por tanto, dependen de su arquitectura/estructura

Ligaduras

- Brazos articulados
  - las ligaduras entre los elementos del robot permiten modelos muy exactos para el control de la posición, orientación y velocidad en un instante t a partir de la posición, orientación y velocidad en un instante t_0
- Robots móviles
  - las ligaduras son menos estrictas y los modelos matemáticos no garantizan el control exacto de la posición, orientación y velocidad en un instante t a partir de la posición, orientación y velocidad en un instante t_0

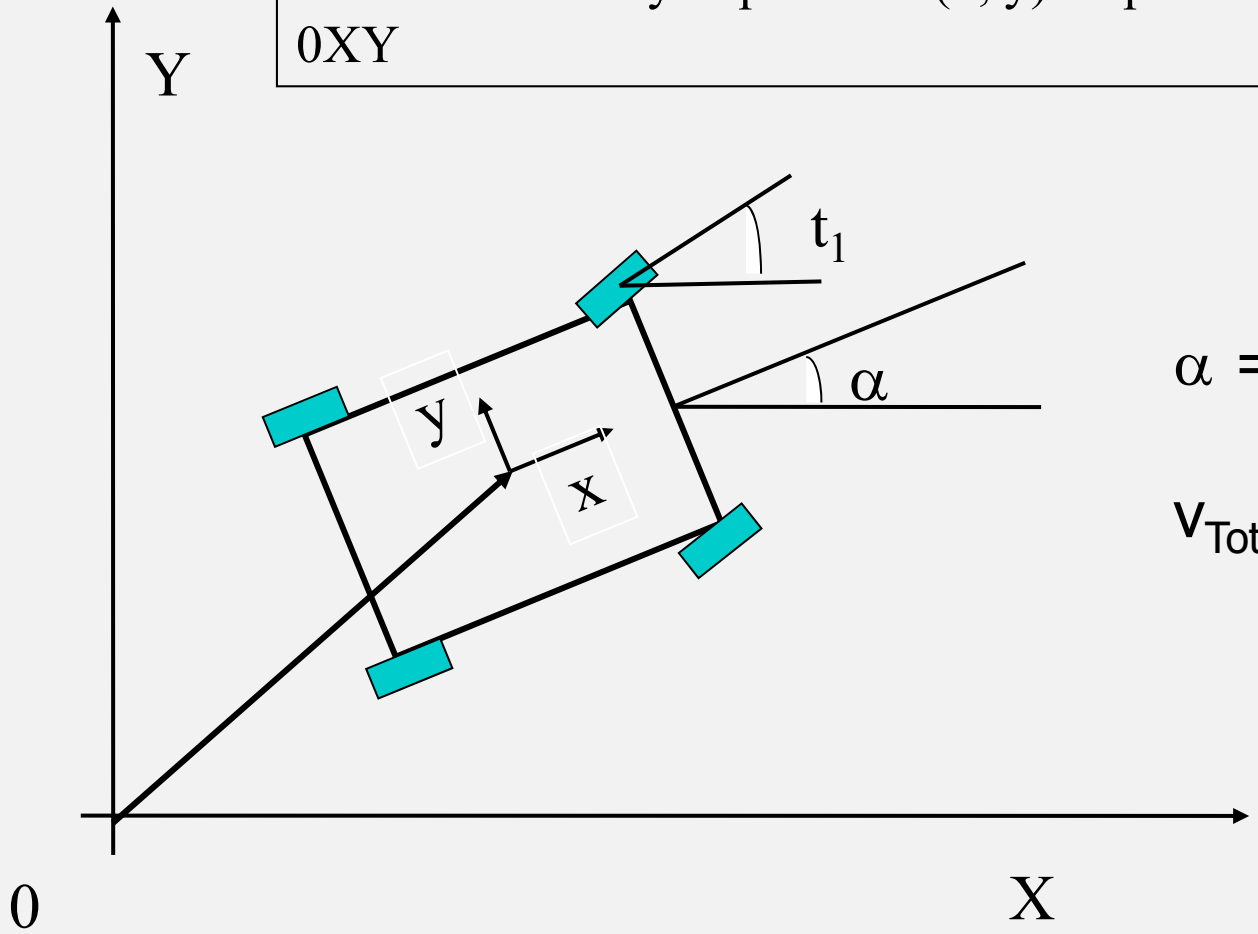
Parámetros espaciales

- Para metros espaciales: describen un objeto en el plano:
  - Un vector de posición P(x,y) respecto del sistema de referencia 0XY
  - El ángulo de giro α del sistema móvil 0X'Y' asociado al objeto, respecto del sistema de referencia fijo 0XY
- Parámetros articulares: describen las posiciones de las articulaciones (t_1, t_2,… t_n)
  - Dependen del tipo de robot: ángulo girado por el eje, ángulo del plano de la rueda, etc.



Cinemática del robot móvil

Dados los ángulos t_1, t_2 (o las velocidades de las ruedas v_R y v_L) calcular la orientación α y la posición (x, y) respecto del sistema de coordenadas 0XY



$$\alpha = (v_R - v_L)/d$$

$$v_{Tot} = (v_R + v_L)/2$$

Problemas básicos

- Problema cinemático directo
  - Dados todos los parámetros articulares $t_1, t_2, \ldots t_n$ (p.e. ángulos de las ruedas, dirección, otras articulaciones, etc.) determinar la posición (x,y) y orientación α del robot

$$x = f_x(t_1, t_2, \ldots t_n)$$
$$y = f_y(t_1, t_2, \ldots t_n)$$
$$\alpha = f_\alpha(t_1, t_2, \ldots t_n)$$

- Problema cinemático inverso
  - Dada la posición (x,y) y orientación (α) del robot, determinar los valores de los parámetros articulares $t_1, t_2, \ldots t_n$ (p. ej. ángulos de las ruedas, dirección, otras articulaciones, etc.)

$$t_1 = f_{t1}(x, y, \alpha)$$
$$t_2 = f_{t2}(x, y, \alpha)$$
$$\ldots..$$
$$t_n = f_{tn}(x, y, \alpha)$$

Problemas con los sistemas reales

Frecuentemente, los robots móviles reales no satisfacen los modelos matemáticos porque:

- La masa:
  - no está uniformemente distribuida en el espacio y puede cambiar con el tiempo
- Contexto físico cambiante
  - puede haber problemas de fricción y deslizamiento
  - Cambios de temperatura, iluminación, interferencias
- Consistencia/Permanencia
  - Las estructuras no son completamente rígidas (las cubiertas se deforman, las correas se estiran, etc.)

Navegación en robótica móvil

Objetivo de la navegación

- Un robot móvil debe ser capaz de ir desde su posición actual A a un destino predefinido B de manera autónoma.
- Para ello, tiene que
  - identificar su posición actual en el mapa
  - encontrar el mejor camino hasta el destino y planificar una ruta adecuada
  - seguir la ruta planeada
  - evitar obstáculos en su camino

- La navegación es el conjunto de procesos orientados a encontrar un camino que une un punto de origen con un punto destino y su correcto seguimiento sin colisiones
  - Las preguntas fundamentales para un robot móvil son:
    - ¿dónde estoy?
    - ¿dónde están las demás cosas respecto a mí?
    - ¿cómo puedo llegar a ellas desde aquí?

Tareas

Mapping

- Adquirir información del entorno
- Posicionar el robot en el entorno

Path planning

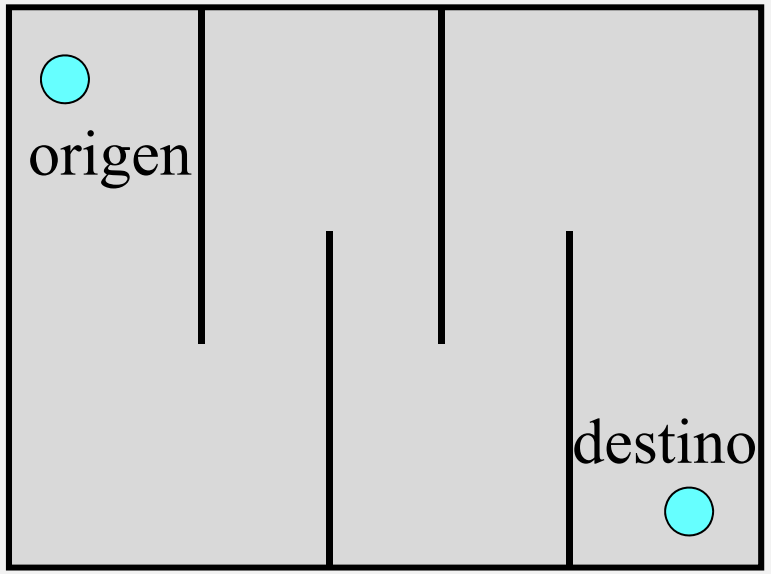
- Determinar caminos
- Elegir el mejor camino

Driving

- Seguir el camino elegido
- Evitar colisiones

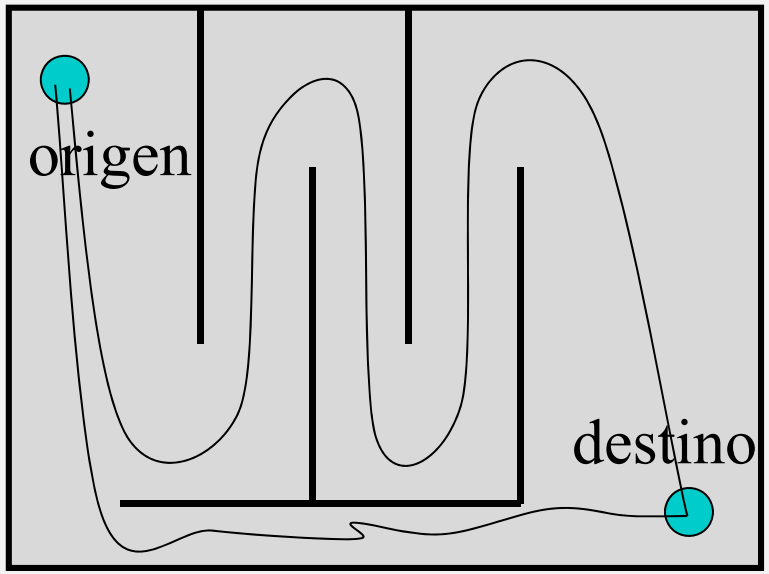
Mapping

- Conocimiento del entorno (=> mapa => modelo)
- Adquisición del mapa:
  - previa a la misión o
  - mientras navega
- Cambios entorno
  - Todo lo que no aparece en el mapa
- Posicionamiento:
  - sensores



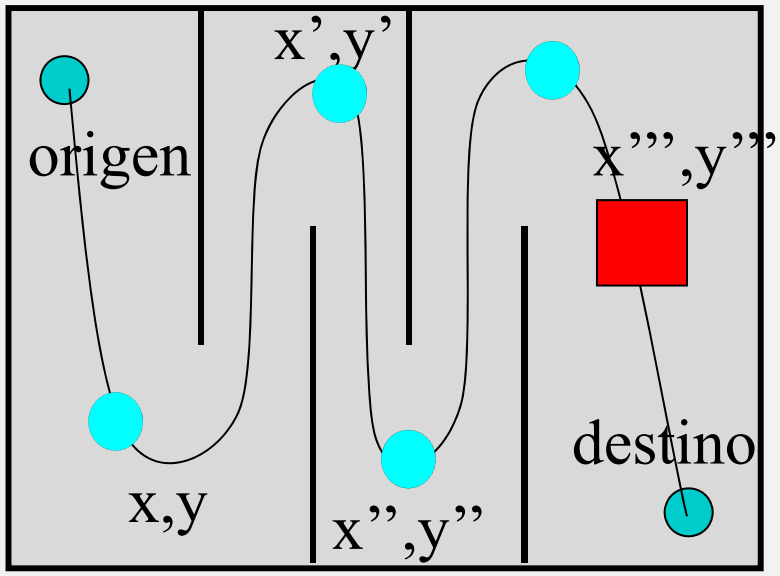
Path planning

- Determinar si existe uno o más caminos practicables
- Si hay más de uno, determinar cuál es el mejor
  - Criterio de selección (distancia, tráfico, pendiente, etc.)



Driving

- Seguir el camino elegido
- Posicionar continuamente el robot
- Detectar y evitar obstáculos



Mapas usados para mapping

Mapas

- Un mapa es una estructura de datos con información válida para la navegación
  - La información espacial contenida en el mapa debe ser coherente con el tipo de información que el robot maneja para identificar las posiciones espaciales.
  - Representa parcialmente el entorno del robot en el ordenador.
  - Dos enfoques complementarios:
    - representaciones continuas (mapas métricos) y
    - discretas (mapas topológicos).

Parámetros y características de los mapas

- Precisión de la representación
  - Compromiso entre requisitos y eficiencia:
    más precisión => más volumen de datos => mayor necesidad de cómputo
- Nº de elementos para representar objetos complejos
- Abstracción:
  - Minimizado de elementos no homogéneos
  - Pocos elementos para las zonas homogéneas
- Compatibilidad con los sistemas planificadores y de localización
- Información complementaria útil para la navegación

Mapas métricos

- Parecidos a los mapas que usan las personas
- Representan el espacio
  - a escala
  - en dos dimensiones
  - Sitúan la información relevante dentro de cada celda (mediante coordenadas precisas)
- Son intuitivos para las personas, pero
  - su construcción es trabajosa y
  - su uso dificulta el cálculo de las distancias
- La representation discreta típica
  - divide el espacio en trozos (cuadriculas o rejillas)
  - de igual o diferentes tamaños
- Las cuadrículas pueden tener diversas resoluciones:
  - más fina donde se necesita más precisión y
  - más gruesa donde el mapa es uniforme

- Ventajas e inconvenientes

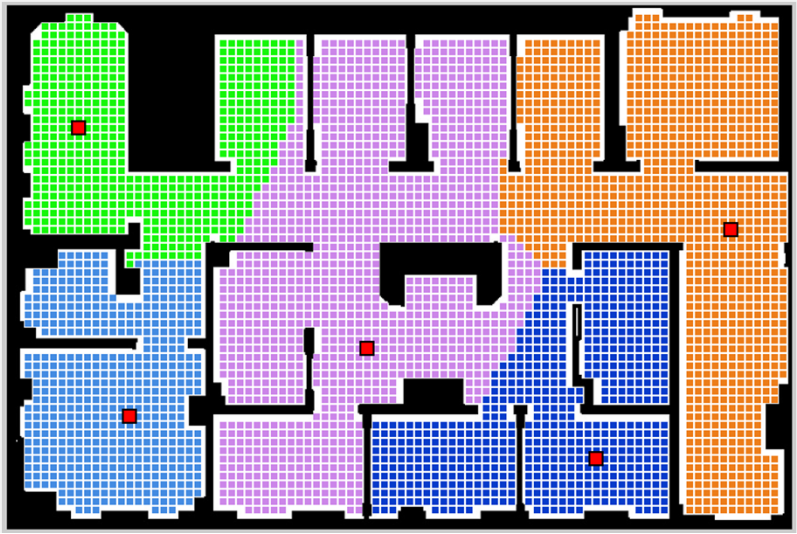
  \+ construcción directa

  \+ aprendizaje y mantenimiento fácil

  \- gran volumen de datos

  \- resolución fija

  \- complejidad de planificación

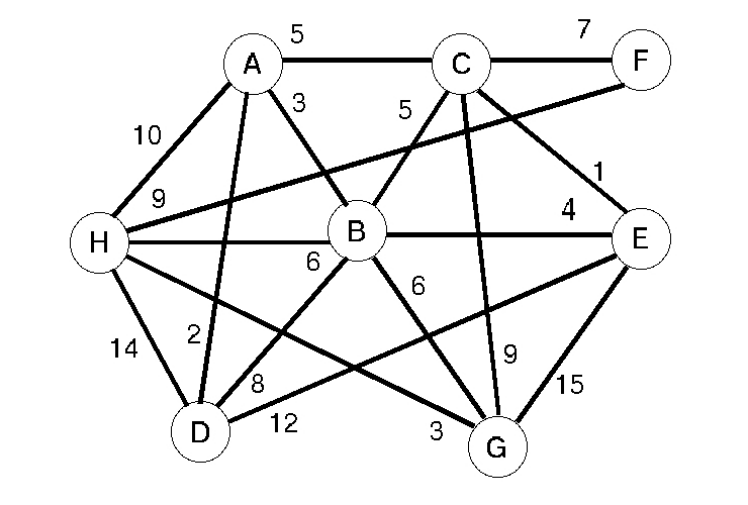


Mapas topológicos

- Sólo tienen en cuenta
  - determinados elementos específicos del espacio y
  - las relaciones entre ellos
- Representación relacional mediante grafos:
  - Cada punto relevante del espacio se representa como un vértice o nodo del grafo
  - Los arcos son caminos entre los lugares
    - Los nodos están conectados por un arco si un robot puede navegar de uno al otro
  - Se puede asignar peso a cada arco siguiendo un determinado criterio (distancia, pendiente, dificultad, etc.).
- Computacionalmente, un grafo puede ser almacenado como una lista o matriz de adyacencia

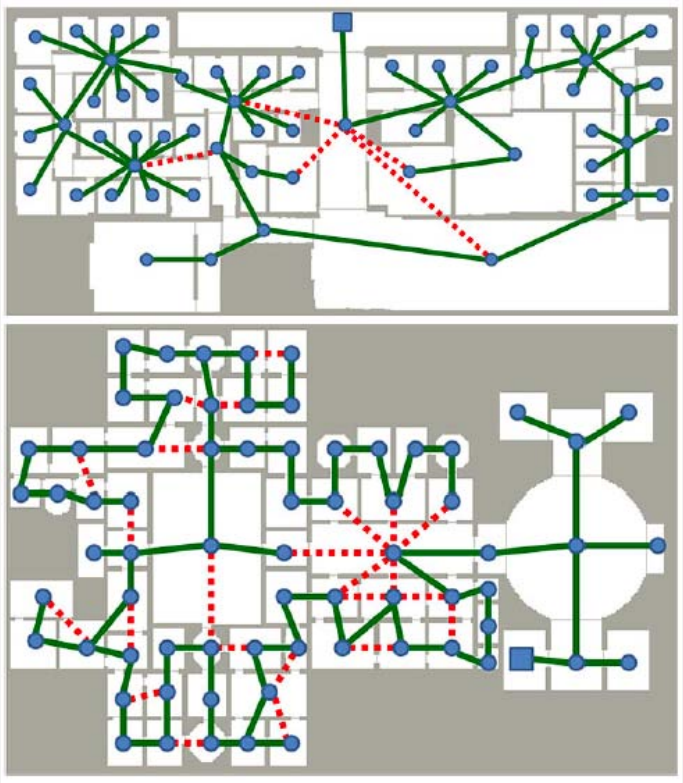
- Ventajas e inconvenientes
  + más compactos
  + facilidad de planificación
  - elaboración más laboriosa

Información relacional (grafos)



Mapas mixtos

- Desventajas de los mapas métricos
  - enormes requisitos de memoria
  - tiempo de cálculo requerido para recorrer estructuras de datos que contienen un gran número de vértices.
- Solución
  - usar mapas topológicos que codifican espacios enteros (ej. habitaciones) como vértices y
  - utilizan aristas para indicar conexiones navegables entre ellos.
  - cada espacio dispone de un mapa métrico detallado a la escala que se a necesaria para navegar en ese lugar.

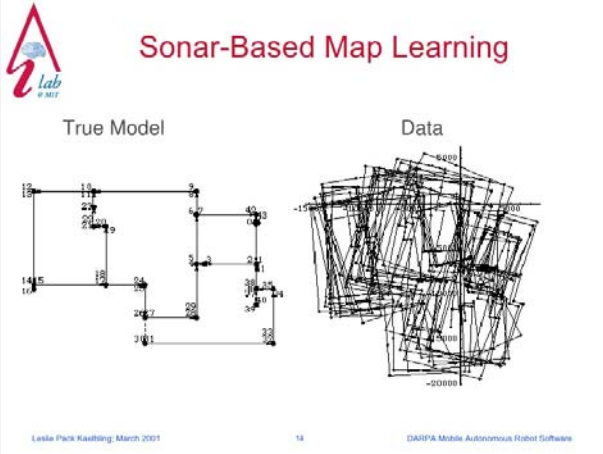


Adquisición del mapa

- Adquisición del mapa por programa
  - Introducido por el operador
- Adquisición del mapa por aprendizaje
  - Simultaneous localization and mapping (SLAM) [Localización y mapeado simultáneo]
    - construir o actualizar un mapa de un entorno desconocido, mientras se registra la ubicación del agente que lo realiza.
    - existen algoritmos para resolverlo en tiempo manejable bajo ciertas condiciones
    - Los métodos más comunes son: filtros de partículas y filtros de Kalman extendidos
  - Los algoritmos SLAM se adaptan a los recursos disponibles y usualmente producen mapas aproximados.

Adquisición del mapa. Ejemplo

- Adquisición a partir de datos de sonar
  - Espacio inicialmente desconocido
  - Robot con **t** transductores de ultrasonidos
  - El robot se desplaza y va midiendo las distancias a los obstáculos
  - Su posición se calcula por odometría
  - D_{ij}= distancia medida por el sensor i en a posición j
  - A cada punto (m,n) de la retícula regular se le asigna $P(m,n)=Sd_{ij}$
- Métodos de corrección
  - Hipótesis restrictivas
    - Orientación perpendicular de las paredes
    - Detección de marcas preestablecidas
  - Mayor número o variedad de datos
    - Fusión de visón, láser, ultrasonidos, sensores inerciales
    - Medidas aportadas por múltiples robots



Mapping: Localización del robot

- El robot tiene que ser capaz de identificar su posición en el espacio y de localizarla en el mapa

- Localización absoluta:
  - determina las coordenadas espaciales (x,y,z) respecto de un sistema de referencia cartesiano 0XYZ
    - Ejemplo: los sistemas GPS devuelven las coordenadas espaciales geográficas (latitud, longitud, altura)
- Localización relativa
  - determina la posición en relación a uno o varios elementos conocidos del mapa
    - Ejemplo: los sensores RFID informan sobre la proximidad a un receptor (específico o genérico)
    - las posiciones de las marcas (naturales o artificiales) situadas en el entorno beben ser localizables en el mapa, para poder deducir la posición del robot

- El uso de un tipo u otro de localización depende de los sensores que tenga el robot

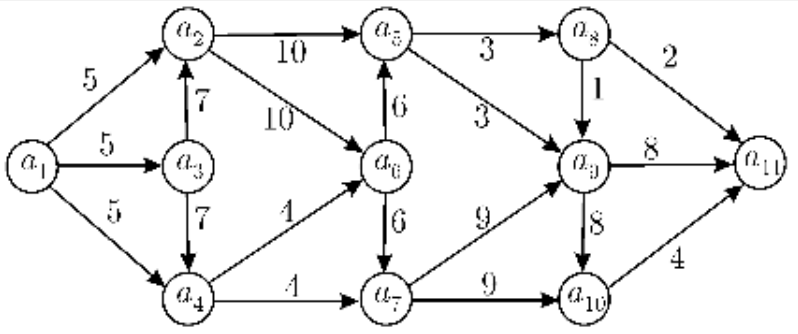
Técnicas básicas de planning

Planificación de caminos

- El tipo de planificador que se usa depende del tipo de mapa disponible
- Fundamentalmente trabajamos con dos tipos de planificadores:
  - Métricos
  - Topológicos

Planificadores topológicos

- Con mapas topológicos, la planificación de la ruta entre dos puntos A y B se reduce a encontrar un camino "optimo" en un grafo conectado.
  - “Optimo” se refiere al coste mínimo acumulativo, que podría ser la distancia, el trafico, la pendiente, etc.
- Grafo => árbol
  - raíz: nodo origen
  - ramas: posibles caminos
- Función de coste: peso de cada arco
- Algoritmos de búsqueda existentes
  - Algoritmo de Dijkstra, Búsqueda BFS, Búsqueda DFS, Kruskal, Floyd Warshall, Ford Fulkerson
  - Compactos, eficientes
  - Coste añadido de generación del grafo y paso al espacio métrico

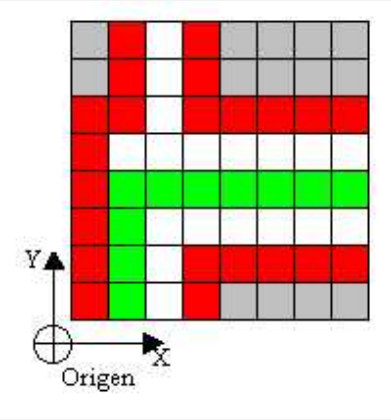


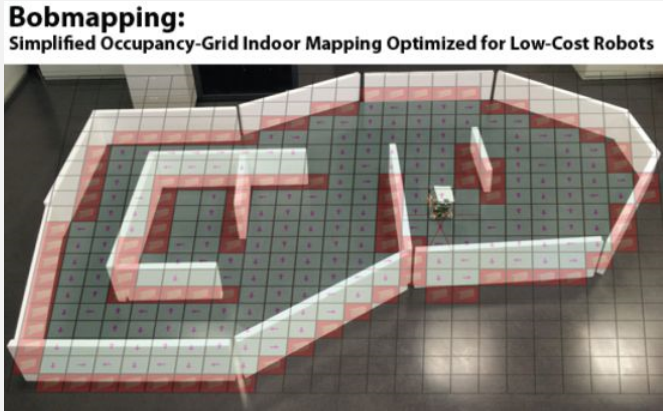
Planificación con mapas topológicos

- Uno de los algoritmos más conocidos y más simples para encontrar la distancia optima entre dos puntos en un grafo es el algoritmo de Dijkstra
  - Partiendo del vértice inicial, el algoritmo de marca todos los vecinos directos del vértice inicial con el costo para llegar a ellos. A continuación, sale del vértice con el menor costo hacia todos sus vértices adyacentes y los marca con el costo para llegar a ellos a través de sí mismo si este costo es menor. Una vez que todos los vecinos de un vértice se han comprobado, el algoritmo continua hasta el vértice con el siguiente costo más bajo. Termina cuando el algoritmo llega al vértice destino B
- Existen algoritmos más eficientes, aunque no óptimos, tales como A*
  - http://es.wikipedia.org/wiki/Algoritmo_de_busqueda_A*
- Una extensión de este algoritmo conocida como D*, aborda el problema de la replanificación cuando aparecen obstáculos en el camino del robot.
  - http://en.wikipedia.org/wiki/D*
  - A diferencia de A*, D comienza desde el vértice objetivo y puede cambiar los costes de los vértices de la ruta que incluyen obstáculos. Esto permite a D* replanificar alrededor de un obstáculo mientras se mantiene la mayor parte del camino ya calculada
- A* y D* se vuelven computacionalmente muy costosos cuando el espacio de búsqueda es grande, por ejemplo, debido al uso de un mapa con una resolución de grano fino

Planificadores métricos

- Estructura de datos que describe el contenido de los elementos de una retícula de n elementos
  - Homogénea
    - todas las casillas son de igual tamaño
  - Heterogénea
    - Los tamaños de las casillas pueden ser diferentes
- Complejidad creciente con n^2
- Inabordable a gran escala
- Adecuados para planificación local
  - Información compatible con datos sensoriales
  - Generación de planes más cortos





Técnicas básicas de navegación usadas en driving

Técnicas básicas de navegación

- Todos los sistemas de navegación suelen incluir tres técnicas de navegación auxiliaries:
  - Guiado
  - Dead Reckoning
  - Evitación de obstáculos

1. Guiado

Tipos de guiado por autonomía/dificultad

- Vehículo autónomo
- Guiado por marcas
- Guiado por balizas
- Guiado estimativo
- Guiado continuo

Grado de dificultad

Guiado continuo

- Seguimiento continuo de un camino preestablecido
- Marcaje de caminos mediante
  - cable soterrado, pintura, campo magnético, etc.
- Aplicaciones:
  - transporte
- Adecuación:
  - entornos controlables y poco cambiantes
- Características:

  \+ Sencillez, control centralizado

  \- Escasa flexibilidad, necesidad de marcaje previo

Guiado mediante balizas

- Requiere un sistema de señalización
  - Activa (envía señales, usualmente electromagnéticas)
  - Posiciones conocidas, generalmente identificables
- Permite la corrección de errores odométricos mediante reposicionado
- Características:

  + Mayor autonomía y fiabilidad

  - Requiere: instalación previa, mapa previo de balizas, mayor capacidad sensorial y de proceso

Guiado mediante marcas

- Requiere un sistema de señalización
  - Pasiva
  - Posiciones conocidas
- Marcas: metas intermedias
  - Artificiales
    - Introducidas para la navegación
    - con o sin identificación)
  - Naturales
    - estaban previamente en el entorno
- Modelo del entorno = grafo
  - Nodos: marcas
  - Arcos: relación entre marcas

Ventajas e inconvenientes.

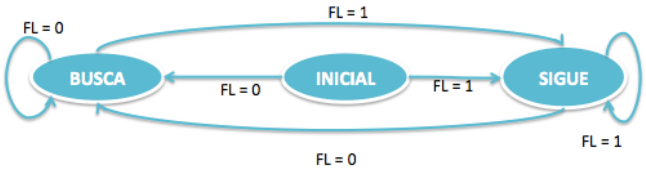
- Marcas artificiales indistinguibles
  + Sencillo: pocos requisitos sensoriales
  - Marcaje previo
  - Falta de referencia absoluta
- Marcas artificiales distinguibles
  + Mayor robustez
  - Marcaje más laborioso
  - Necesidad de mayor capacidad sensorial
- Marcas naturales
  + Sin marcaje previo
  - Necesidad de gran capacidad perceptiva (visión)

Seguimiento de una marca lineal

- Se establece un “carril” (en el suelo, el techo, o la pared) que debe ser seguido por el robot
- El robot debe disponer de sensores capaces de reconocer el carril
  - Cámaras de video para reconocer tubos o esquinas en el techo
  - Sensores mágneticos para reconocer cables eléctricos enterrados en el suelo
  - Senores de luz para reconocer líneas de color pintadas en el suelo
  - etc.

- El algoritmo de control utiliza los datos de los sensors para seguir la línea

L	FL	FR	R	W	ESTADO ANTERIOR	NUEVO ESTADO
-	0	-	-	-	INICIAL	BUSCA: {Girar derecha}
-	1	-	-	-	INICIAL	SIGUE: {Avanzar}
-	0	-	-	-	SIGUE	BUSCA
-	1	-	-	-	SIGUE	SIGUE
-	0	-	-	-	BUSCA	BUSCA
-	1	-	-	-	BUSCA	SIGUE

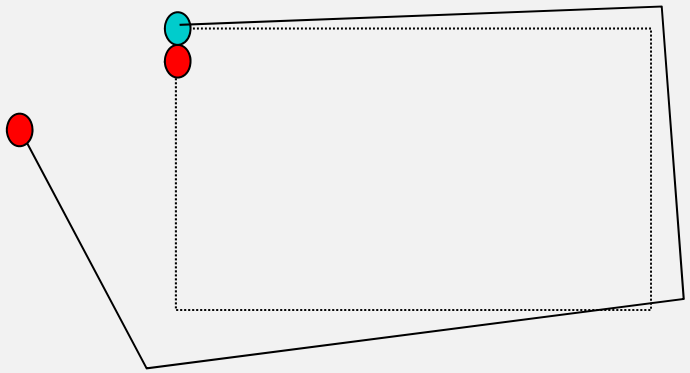


Guiado estimativo

- *Dead reckoning*
  - Pos. actual = pos. de referencia + movimiento estimado
- Sensores:
  - Odometría: usa información sobre la rotación de las ruedas para estimar cambios en la posición a lo largo del tiempo
  - Doppler: velocidad a partir de la diferencia entre las frecuencias de las ondas emitida y reflejada
  - Inerciales: acelerómetros (aceleración lineal), giróscopos (velocidad angular) y magnetómetros (norte magnético)
- Características:

  \+ Definición de caminos por programa

  \- Errores acumulativos



2. Dead Reckoning

- El *dead reckoning* es un tipo de navegación estimativa, asociada a la detección de marcas o balizas
- Consiste en una técnica usada para navegar “a ciegas” entre dos puntos conocidos
  - El robot "estima" los movimientos que debe realizar para llegar al destino previsto
  - Realiza esos movimientos sin disponer de feedback
  - Cuando llega al punto final estimado debe aplicar algún método de localización relativa para corregir los errores que se hayan acumulado en el recorrido (usualmente por medio de marcas o balizas)
- La implementación más simple del dead reckoning se hace por odometría utilizando encoders ópticos directamente acoplados a los motores o a los ejes de las ruedas
  - Además existen multitud de sensores para cuantificar el desplazamiento angular y la velocidad, como por ejemplo: potenciómetros, sincros, resolvers, encoders magnéticos, inductivos y capacitivos, etc.
- El dead reckoning
  - Está presente junto a otros sistemas de navegación
  - Se aplica cuando no se dispone de feedback contextual

Odometría para Dead Reckoning

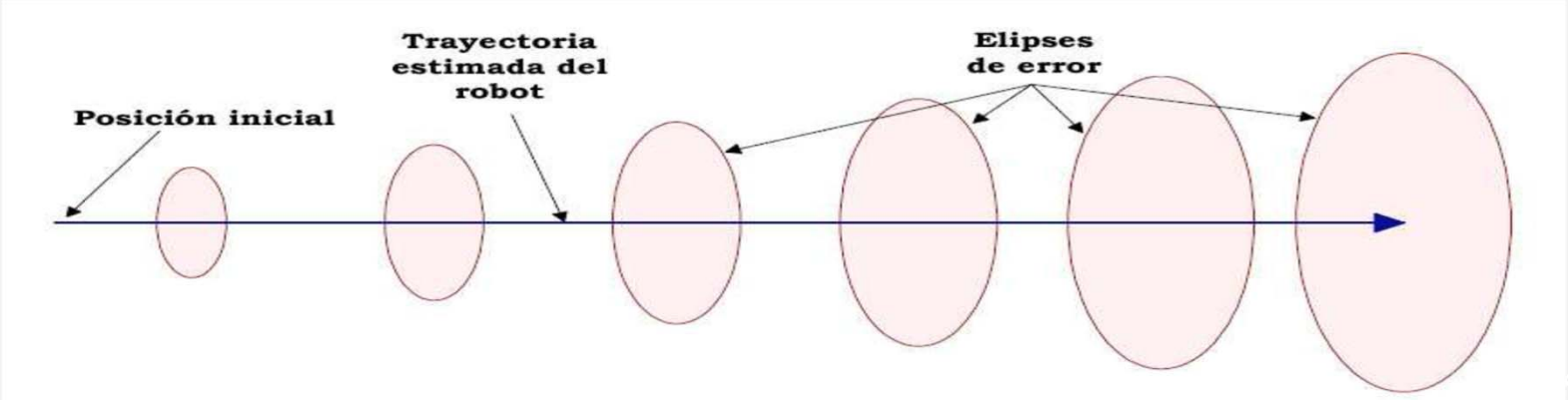
- La ***odometría*** es uno de los métodos más ampliamente usados para estimar la posición de un robot
  - proporciona buena precisión a corto plazo
  - es barata
  - permite tasas de muestro muy altas

- Conlleva una inevitable acumulación de errores
  - se basa en la integración de información incremental del movimiento a lo largo del tiempo
  - la acumulación de errores de orientación, puede causar desviaciones en la estimación de la posición que van aumentando proporcionalmente con la distancia recorrida por el robot

- A pesar de estas limitaciones, la odometría es una parte fundamental del sistema de navegación de la mayoría de los robots

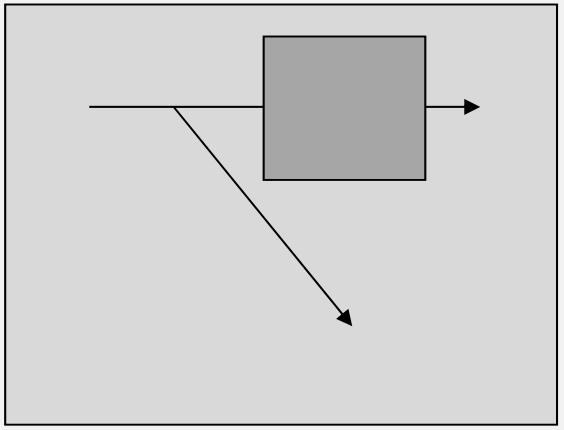
Errores sistemáticos y esporádicos en odometría.

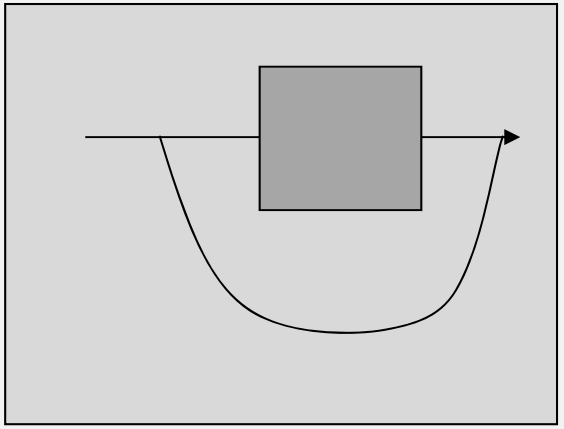
- Los errores sistemáticos se producen siempre:
  - cuando los diámetros de las ruedas no son iguales, las ruedas están mal alineadas, el encoder tiene poca resolución, etc.
  - se acumulan constantemente, pero son predecibles y modelables
- Los errores esporádicos se producen aleatoriamente:
  - por ejemplo, por moverse en suelos desnivelados, o sobre obstáculos, deslizamiento de las ruedas (por falta de agarre, sobre-aceleración, derrape, falta de contacto con el suelo, etc.)
  - son impredecibles, por tanto, no modelables

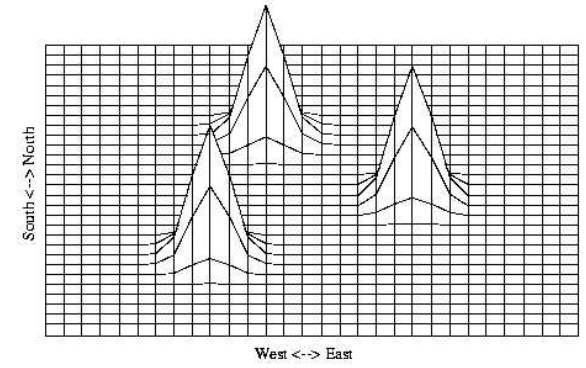


3. Evitación de obstáculos

- Navegación local:
  - De sub-objetivo a sub-objetivo
  - Evitación de obstáculos
  - Operación en tiempo real
- Obstáculos
  - Objetos puestos después de la creación del mapa
  - Zonas del entorno no representadas en el mapa
  - Personas u otros robots en cohabitación
  - Desalineación de sistemas de referencia robot-mapa
- Técnicas de control reactivo
  - Vagar
  - Circunvalar
  - Campo de potencial







Bibliografía

- Gaurav Sukhatme. Introduction to Sensing and Sensor Fusion . USC Robotics Research Laboratory. www-robotics.usc.edu

- Mobile Robots. Inspiration to implementation.AK Peters 1992.

- J-C. Latombe. Robot motion planning. KAP. 1991

Apéndice I
Odometría para navegación por "dead reckoning" en robótica móvil

Dead reckoning, o navegación estimada

- Dead reckoning es una expresión derivada del término náutico *deduced reckoning (estimación deducida)*
- Es una técnica usada para navegar “a ciegas” entre dos puntos conocidos
- Cuando se llega al punto de destino estimado es necesario aplicar algún método de localización relativa para corregir los errores que se hayan acumulado en el recorrido
- La implementación más simple del *dead reckoning* se hace por odometría utilizando encoders ópticos directamente acoplados a los motores o a los ejes de las ruedas

Odometría

- La odometría se utiliza para estimar la posición relativa de un robot móvil según va moviéndose
  - las posiciones estimadas a partir de la odometría son relativas al punto de inicio (y no tienen relación con las coordenadas del mundo)
- Para navegar por odometría es necesario
  - acoplar encoders a los motores para medir el avance de cada rueda (teniendo en cuenta el radio de la misma y la reducción del motor)
  - aplicar las ecuaciones cinemáticas del robot para hallar la posición en la que se encuentra y el ángulo de desviación respecto a una dirección de referencia
  - De este modo se puede calcular cuánto se desplaza linealmente cada rueda (de radio r) por cada pulso registrado por el encoder (de resolución C_e) donde n es la reducción eje/rueda:

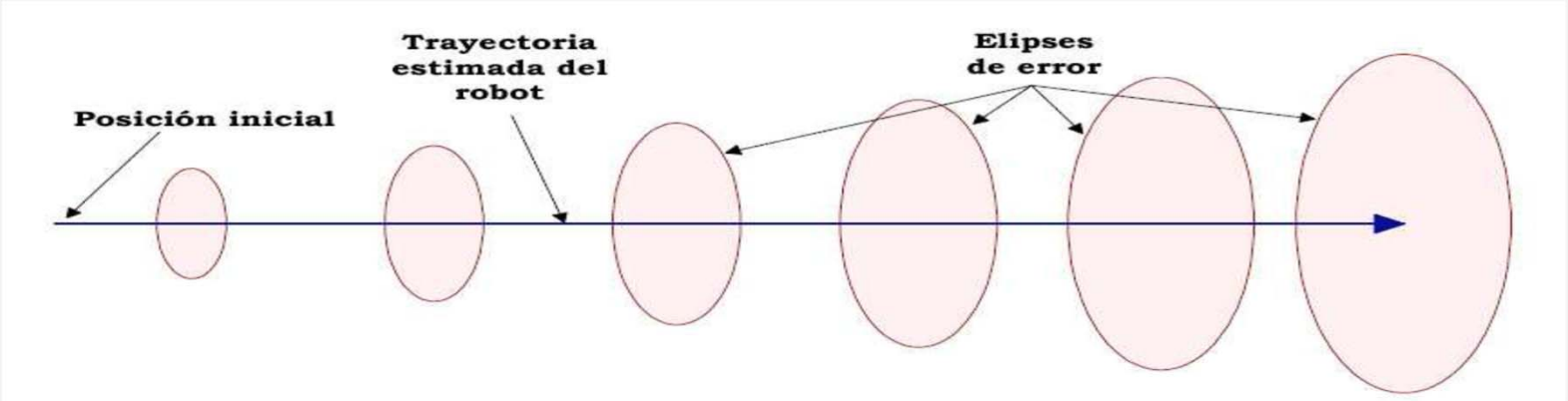
$$c_m = 2\pi r/nC_e$$

Odometria

- La ***odometría*** es uno de los métodos más usados para estimar la posición de un robot
  - buena precisión a corto plazo
  - barata
  - tasas de muestro muy altas
- Sin embargo conlleva la acumulación de errores que van aumentando proporcionalmente con la distancia recorrida

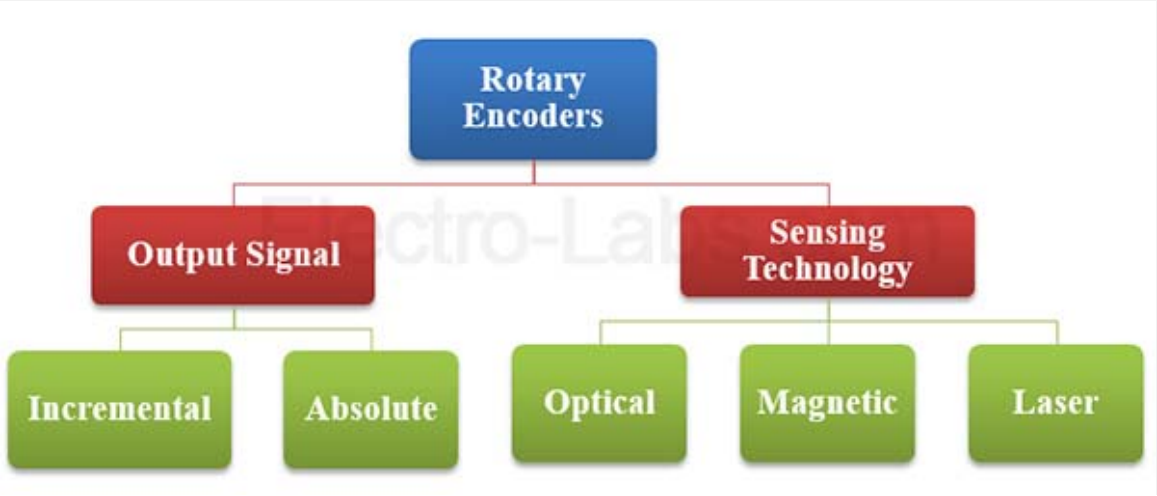
Errores sistemáticos y no sistemáticos en odometría.

- Los errores sistemáticos se producen siempre:
  - cuando los diámetros de las ruedas no son iguales, las ruedas están mal alineadas, el encoder tiene poca resolución, etc.
  - se acumulan constantemente, pero son predecibles y modelables
- Los errores no sistemáticos se producen aleatoriamente:
  - por ejemplo, por moverse en suelos desnivelados, o sobre obstáculos, deslizamiento de las ruedas (por falta de agarre, sobreaceleración, derrape, falta de contacto con el suelo, etc.)
  - son impredecibles, por tanto, no modelables



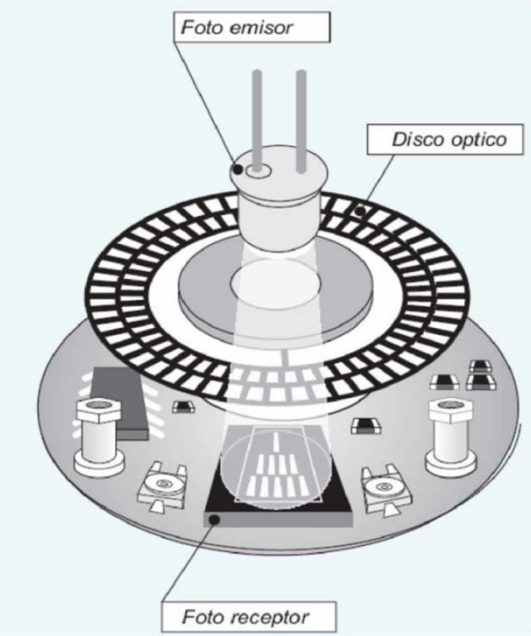
Codificadores rotatorios (rotary encoders)

- Son sensores que generan una señal eléctrica (analógica o digital) en respuesta a un movimiento de rotación
  - Sirven para determinar (y controlar) la posición o velocidad de un dispositivo mecánico
  - Suelen ir montados sobre un eje cilíndrico
  - Se utilizan en diversas aplicaciones para el control de velocidad y/o posición
- Tipos de codificadores rotatorios



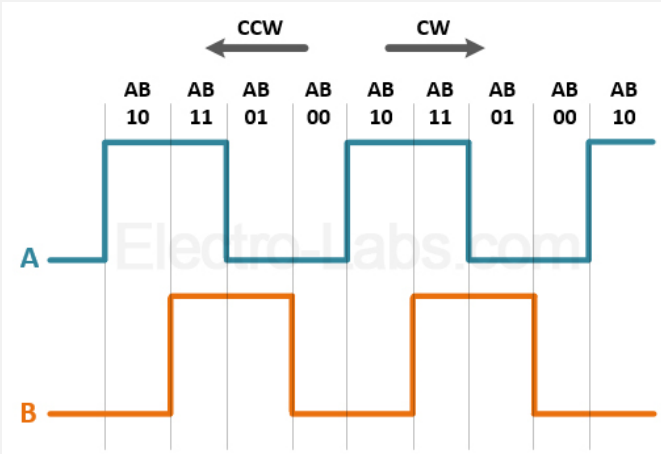
Sensores para odometría: Encoders ópticos

- Un encoder está formado por un disco con ranuras radiales ubicadas en toda su circunferencia que giran frente a un fotosensor generando un pulso por cada ranura.



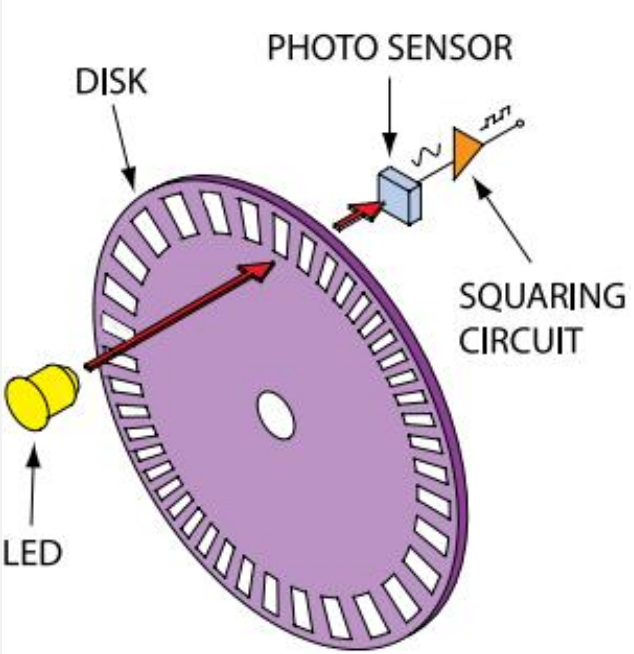
Encoders incrementales o relativos

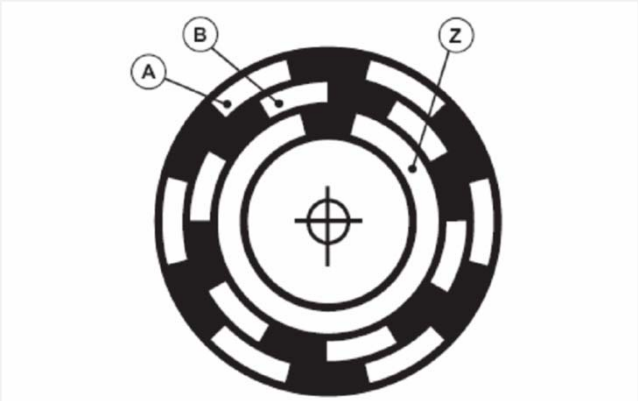
- El encoder incremental proporciona dos formas de onda cuadradas y desfasadas 90º entre sí:
  - un canal da información de la velocidad de rotación
  - el otro permite discriminar el sentido de rotación comparando las secuencias



Encoders incrementales o relativos

- Además puede haber otra señal llamada canal Z (cero), que proporciona la posición absoluta de cero del eje del encoder.
  - Esta marca indica que se ha dado una vuelta completa
  - También sirve para inicializar la cuenta después de haya ocurrido algún error.

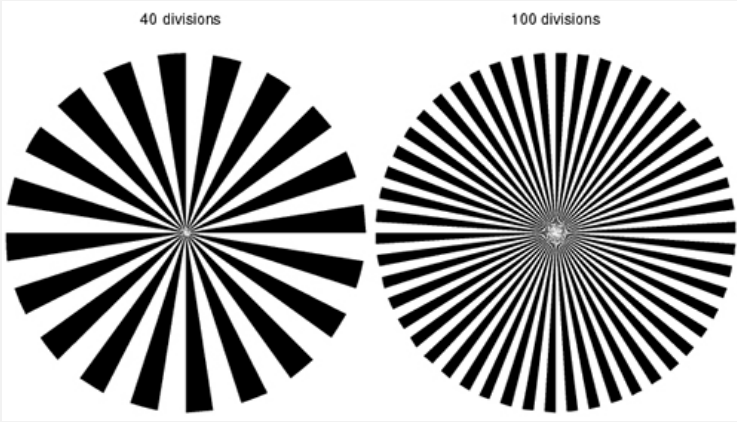




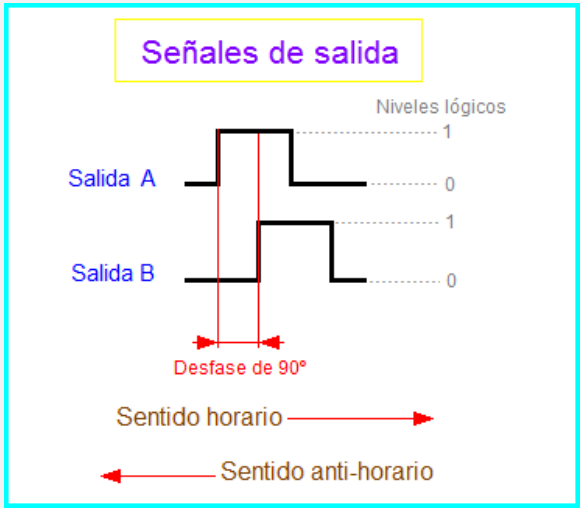
Representación de las señales incrementales A, B y Z en disco óptico.

Encoders incrementales o relativos

- La resolución del dispositivo viene dada por el número de segmentos.
  - 40 divisiones -> 20 franjas transparentes -> 20 pulsos por vuelta.
  - Si tiene dos canales podemos distinguir 4 partes en cada pulso (4 códigos 00, 10, 11,01) - > Resolución: 360º/80 = 4.5º/código
  - Por ejemplo, si contamos 16 códigos:
    $\alpha = 16*4.5º \pm 4.5º = 72º \pm 4.5º$
    $\alpha = [67.5 – 76.5]º$



- Ventajas:
  - simplicidad
- Desventajas:
  - requieren contadores externos para determinar el ángulo y la dirección de rotación del eje

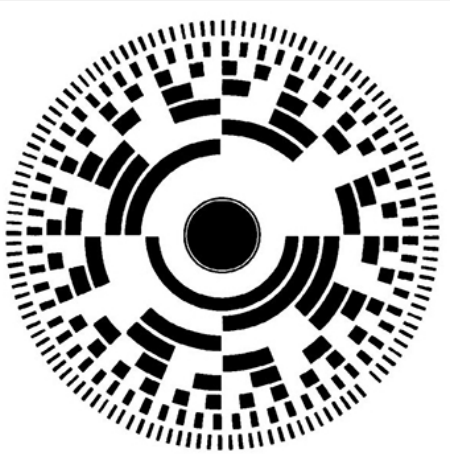


Encoders absolutos

- El funcionamiento físico es similar al de los encoders incrementales
  - El disco transparente se divide en **p** de pistas cada una de las cuales corresponde a un bit del código de posición
  - Cada pista está dividida en un **s** sectores (blancos y negros) tal que

$$p = \log_2 (s),$$

    - Para que p sea entero n debe ser potencia de 2
  - De esta manera, proporcionan un código binario único para cada posición de rotación
  - Los códigos se repiten en cada revolución
  - No es necesaria ninguna señal adicional para detectar el sentido de giro



Codificadores o encoders absolutos

- La resolución viene dada por el número de pistas **p** en el disco graduado
  - Cada pista representa un bit del código
  - Se requieren **$s = 2^p$** sectores para codificar todos los códigos posibles
- Por ejemplo:
  - Con 8 bits: $2^8 = 256$ sectores codifican 256 posiciones [precisión: 360º/256 = 1.4º/código]

    Por ejemplo, si detectamos un giro de 16 códigos:

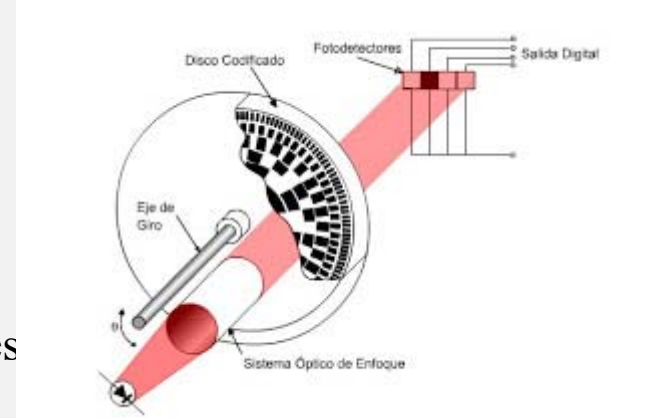
    $\alpha = 16*1.4º \pm 1.4º = 22.4º \pm 1.4º$

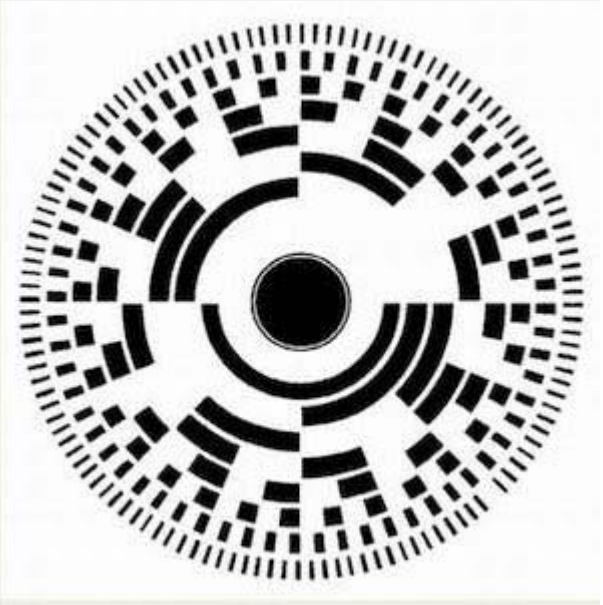
    $\alpha = [21.0 – 23.8]º$
  - Con 19 bits: $2^{19} = 24.288$ sectores codifican 24.288 posiciones [precisión: 360º/24.288= 0.015º/posición]).

    Por ejemplo, si detectamos un giro de 16 códigos:

    $\alpha = 16*0.015º \pm 0.015º = 0.24º \pm 0.015º$

    $\alpha = [0.235 – 0.255]º$





Encoders absolutos

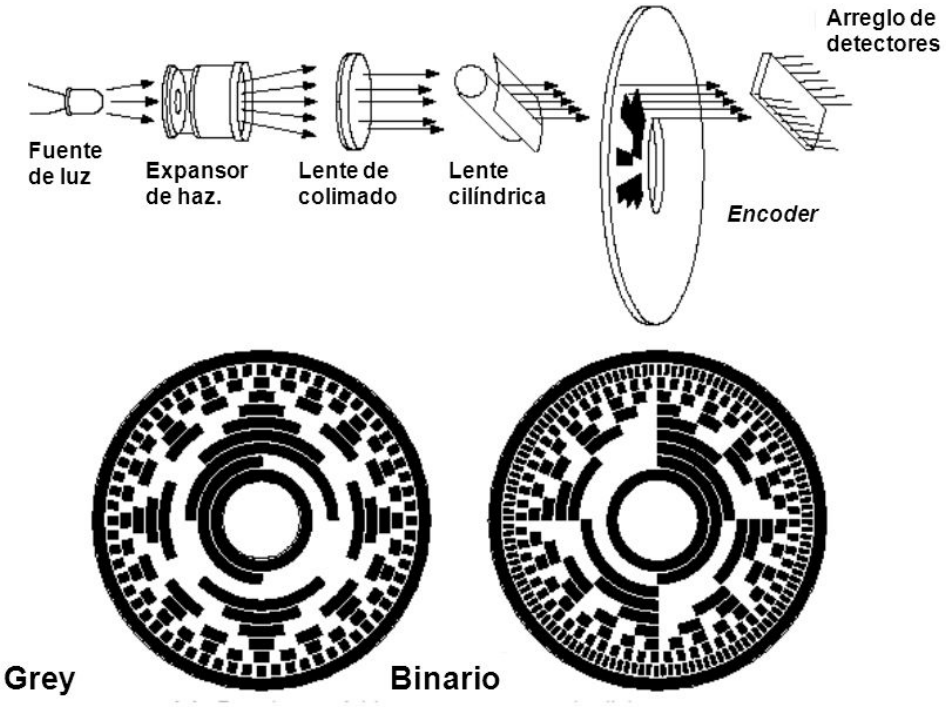
- Ventajas:
  - conservarla posición exacta, incluso en caso de fallo de alimentación
  - no requieren un contador exterior
- Inconvenientes:
  - son más complejos y costosos que los codificadores incrementales

Codificadores o encoders absolutos

- Para evitar los errores producidos en cambios no simultáneos en uno de los bits del código se usa un código binario cíclico (Gray) que sólo cambia un bit por transición

Decimal	Código binario	Código de Gray
0	0000	0000
1	0001	0001
2	0010	0011
3	0011	0010
4	0100	0110
5	0101	0111
6	0110	0101
7	0111	0100
8	1000	1100
9	1001	1101
10	1010	1111
11	1011	1110
12	1100	1010
13	1101	1011
14	1110	1001
15	1111	1000

Sensor para un *encoder* absoluto



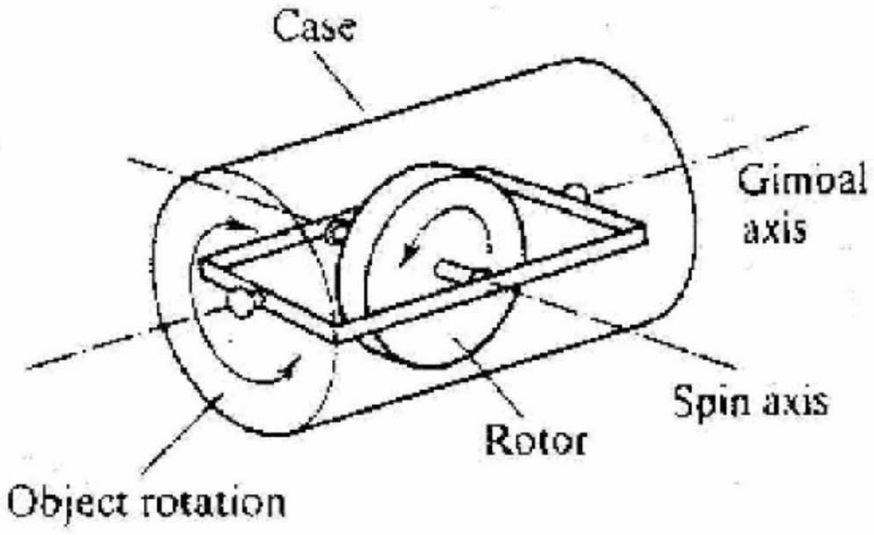
Ejercicio

Queremos hacer un encoder óptico con 7 pistas concéntricas ¿Cuántos sectores por pista requiere? ¿De cuántos radianes es cada sector? ¿Este encoder permitirá detectar la posición absoluta? ¿Permitirá detectar la dirección del movimiento? ¿Por qué?

Apéndice II
Otros sensores para navegación

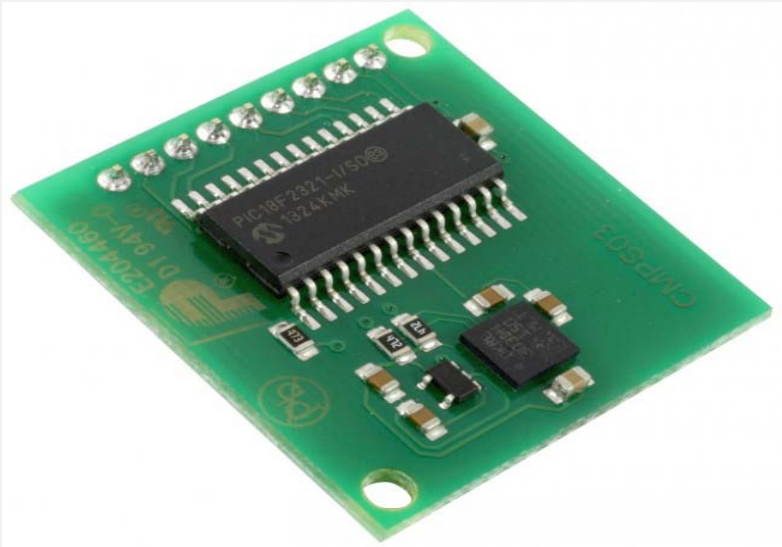
Giróscopos

- Se basa en el *principio de conservación del momen*
  - un cuerpo que gira presenta fuerzas que se oponen al camb lo mismo, la dirección del eje de rotación).
  - Giróscopo digital comercial ROBOTIS Gyro Sensor GS-1



Sensores magnéticos

- La brújula, o *compass*, aplica la detección del campos magnético de la tierra.
- Útil como parte del sistema de orientación cuando el robot se mueve en exteriores.
- Brújula digital comercial: CMPS03 - Compass Module.

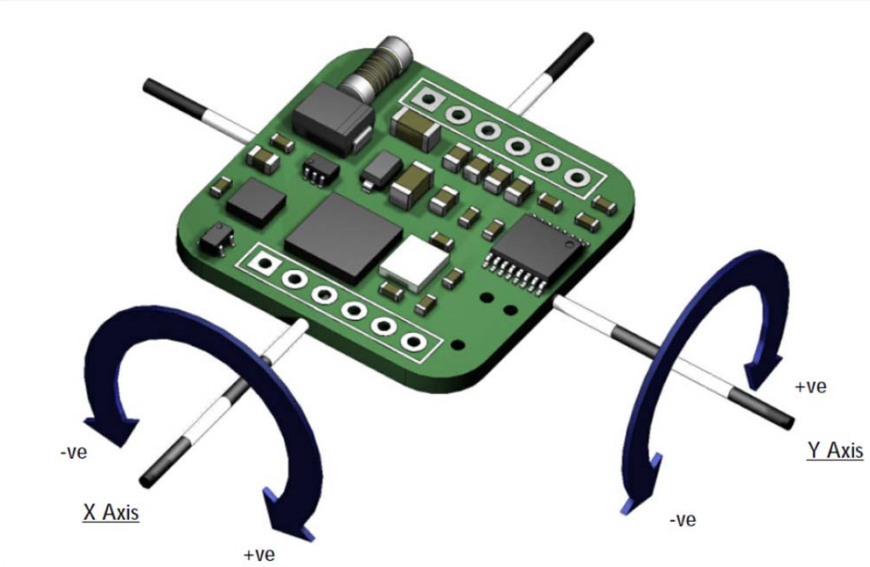


- Sensores magnéticos para medición de movimientos:
  - detectores de "movimiento cero"
  - tacómetros basados en sensores magnéticos que usan el efecto Hall
- Medición de corrientes en la parte motriz (detectando la intensidad del campo magnético que genera un conductor en la fuente de alimentación)

Inclinómetros

- Los inclinómetros sirven para medir el ángulo de un objeto con respecto a un eje horizontal.
- Están formados por un electrolito (liquido conductor) situado en un recipiente en el cual hay introducidos dos electrodos.

  - Cuando el sensor se inclina, uno de los electrodos entra más en contacto con el electrolito y el otro menos. Si se comparan las corrientes de salida de los electrodos es posible determinar el ángulo de inclinación.
  - Inclinómetro digital comercial LCP-45.



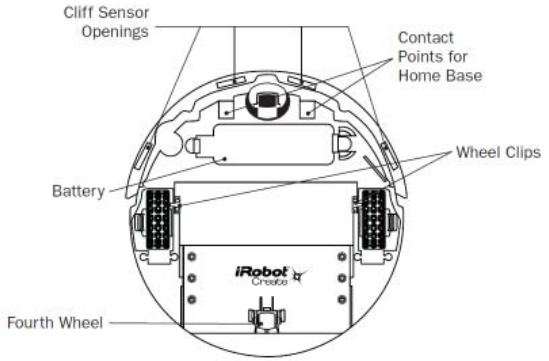
Apéndice III
Navegación por *dead reckonin*g con una configuración diferencial

Odometría con iRobot Create

- Queremos saber la posición (x,y) y orientación (θ) en el plano de un robot con configuración diferencial

$$P(x,y,\theta)$$

- Usaremos los encoders del iRobot Create.
  - son sensores (propioceptivos) que permiten contar las revoluciones del eje de cada rueda
  - El iRobot create es un robot móvil de configuración diferencial:
    - Cuenta con dos ruedas motrices independientes.
    - Cada una de ellas ruedas tiene su propio motor que la hace girar en el plano perpendicular a su eje.



iRobot Create Bottom View

Odometría con iRobot Create

- El iRobot tiene una configuración diferencial con dos ruedas motrices independientes.
- Cada una de ellas tiene su propio motor y un encoder que mide las revoluciones de su eje.
- Las ruedas delantera y trasera son ruedas pasivas (o locas) que se utilizan para dar estabilidad al robot y tienen motores ni encoders.
- La resolución de los encoders se mide en pulsos/revolución.
  - Un encoder de resolución R genera r pulsos en cada vuelta completa del eje.
  - r permite hacer una estimación discreta del ángulo girado por la rueda.
  - Por ejemplo, si el encoder tiene una precisión de 360 pulsos/revolución, su resolución angular será: R= 360 pulsos/360 º ; R = 1º/pulso

Ecuaciones para estimar la posición en una base diferencial

- Queremos estimar la posición **(x,y,θ)** en el plano de un robot diferencial mediante la información dada por los encoders.
- Necesitamos saber cuánto se desplaza linealmente cada rueda por cada pulso registrado por el encoder.
- La ecuación que permite calcular el factor de conversión "pulso de encoder/desplazamiento lineal" es:

$$c_m = 2\pi R/nC_e$$

  donde R es el radio de la rueda, n es la reducción eje/rueda y C_e es la resolución del encoder.
- Los sistemas rueda/motor suelen tener una reducción que hace que una vuelta da la rueda requiera n vueltas del eje del motor.
  - Nosotros consideraremos que n = 1, para simplificar (Es decir cada vuelta que da el eje del motor, la rueda da una vuelta completa, lo que no es en absoluto común).
- Conociendo el factor de conversión $\mathbf{c_m}$ se puede calcular el desplazamiento lineal de cada rueda en un intervalo de tiempo **i**.
- Supongamos que los encoders generan pulsos en intervalos regulares de tiempo.
  - Llamaremos $\mathbf{N_{L,i}}$ al número de pulsos generados por el encoder de la rueda izquierda para el tiempo de muestreo **i**.
  - Igualmente, $\mathbf{N_{R,i,}}$ será el número de pulsos generados por el encoder de la rueda derecha en el intervalo **i**.
- Así pues, la distancia avanzada por cada rueda en el intervalo de tiempo i viene dada por las ecuaciones:

$$\Delta S_{L,i} = C_m N_{L,i}$$
$$\Delta S_{R,i} = C_m N_{R,i}$$

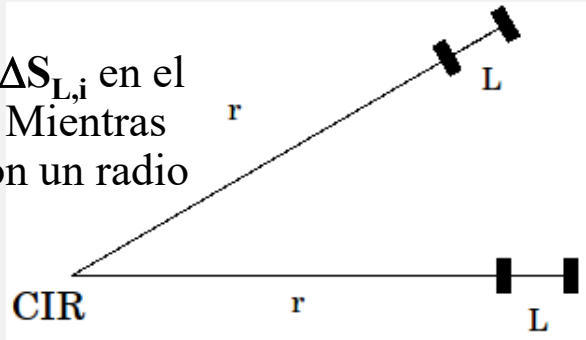
Ecuaciones para estimar la posición en una base diferencial

- $\Delta S_{c,i}$ denota cuánto ha avanzado el centro del robot en ese intervalo de tiempo
- $\Delta \theta_{c,i}$ denota el cambio de orientación que ha experimentado
- Aplicamos las ecuaciones:

$$\Delta S_{c,i} = (\Delta S_{R,i} + \Delta SL,i)/2$$
$$\Delta \theta c,i = (\Delta S_{R,i} - \Delta S_{L,i})/L$$

donde **L** es la distancia entre las dos ruedas.

- Supongamos que cada una de las ruedas lleva una velocidad angular constante $\omega_R = k \neq \omega_L = k'$
  - $\omega_R - \omega_L$ hace que el robot se mueva en una trayectoria curva.
  - Esa trayectoria corresponderá a una circunferencia de radio r, en torno al Centro Instantáneo de Rotación o CIR, que está en la línea que une ambas ruedas
  - Por lo tanto, la rueda izquierda describe un arco de longitud $\Delta S_{L,i}$ en el intervalo **i**, con un radio de curvatura (la distancia al CIR) **r**. Mientras tanto, la rueda derecha describe un arco de longitud $\Delta S_{R,i}$ con un radio **r+L**.



Ecuaciones para estimar la posición en una base diferencial

- El cambio en la orientación $\Delta\theta$ se puede deducir:

$$\Delta S_{L,i}/r = \Delta\theta_i$$
$$\Delta S_{R,i}/(r+L) = \Delta\theta_i$$

- Resolviendo el sistema:

$$\Delta\theta_i = (\Delta S_{R,i} - \Delta S_{L,i})/L$$
$$r = L\ \Delta S_{L,i}/(\Delta S_{R,i} - \Delta S_{L,i})$$

- Para calcular el desplazamiento lineal ΔS_c del centro del robot tenemos en cuenta que describe un arco de longitud $\Delta S_{c,i}$ con radio $\mathbf{r + L/2}$. S
- Sabemos que $\Delta S_{C,i} / (r+L/2) = \Delta\theta_i$. De donde obtenemos

$$\Delta\theta_i = (\Delta S_{R,i} - \Delta S_{L,i})/L$$

- A partir de estas ecuaciones deducimos que

$$\Delta S_{C,i} = (\Delta S_{R,i} - \Delta S_{L,i})/2$$

- Para calcular el vector (x,y,θ) del centro del robot en el instante i suponemos que conocemos las coordenadas del robot en el instante i-1: $(x_{i-1}, y_{i-1}, \theta_{i-1})$.

$$\theta_i = \theta_{i-1} + \Delta\theta_i$$
$$x_i = x_{i-1} + \Delta S_{c,i} \cos\theta_i$$
$$y_i = x_{i-1} + \Delta S_{c,i}\ sen\,\theta_i$$

- Este proceso iterativo requiere conocer las coordenadas del origen del movimiento (x_0, y_0, θ_0). Lo más común es suponer que se parte de (0.0.0) y hacer una estimación de la posición relativa al punto de comienzo.