

RESUMEN TEORÍA ROBÓTICA MÓVIL:

ÍNDICE DE CONTENIDOS:

TEMA 1: INTRODUCCIÓN

- 1.1 CONCEPTOS BÁSICOS**
- 1.2 ARQUITECTURA DE UN ROBOT**
- 1.3 SENSORES**
- 1.4 CÁMARAS**
- 1.5 ACTUADORES**
- 1.6 OTROS ELEMENTOS Y MÓDULOS**
- 1.7 LOCOMOCIÓN Y NAVEGACIÓN**

TEMA 2: SISTEMAS REACTIVOS Y DE CONTROL

- 2.1 COMPORTAMIENTO DE UN ROBOT**
- 2.2 SISTEMAS REACTIVOS**
- 2.3 AUTÓMATAS DE ESTADO FINITO**
- 2.4 TEORÍA DE CONTROL CLÁSICA**
- 2.5 TIPOS DE CONTROLADORES**

TEMA 3: NAVEGACIÓN DE ROBOTS MÓVILES

- 3.1 NAVEGACIÓN**
- 3.2 NAVEGACIÓN LOCAL**
- 3.3 TÉCNICAS DE NAVEGACIÓN LOCAL**
- 3.4 NAVEGACIÓN GLOBAL**
- 3.5 TÉCNICAS DE NAVEGACIÓN GLOBAL**
- 3.6 NAVEGACIÓN HÍBRIDA Y DE ESTÍMULO**

TEMA 4: MAPAS

- 4.1 MAPAS**
- 4.2 TIPOS DE MAPAS**
- 4.3 TÉCNICAS DE CONSTRUCCIÓN DE MAPAS**
- 4.4 APROXIMACIÓN PROBABILÍSTICA**
- 4.5 APROXIMACIÓN BORROSA**

TEMA 5: AUTOLOCALIZACIÓN DE ROBOTS

- 5.1 LOCALIZACIÓN**
- 5.2 ODOMETRÍA**
- 5.3 GPS (TIPOS Y ERRORES)**
- 5.4 BALIZACIÓN**
- 5.5 LOCALIZACIÓN PROBABILÍSTICA**
- 5.6 MODELOS EN LOCALIZACIÓN**
- 5.7 AUTOLOCALIZACIÓN 3D**

TEMA 1: INTRODUCCIÓN

1.1 CONCEPTOS BÁSICOS

Software de un robot: Determina el comportamiento del robot y establece cómo se coordinan percepción y actuación, poseen requisitos específicos (vivacidad, agilidad y robustez).

Software de un robot (características): Rápido, multitarea y multihilo, distribuido en piezas de software que se comunican unas con otras, suelen tener interfaz gráfica sólo para depuración y para que su software sea expandible, conectado a la realidad física.

Sistemas operativos y plataformas: Middleware que simplifica la creación de apps robóticas, pueden ser dedicados o generalistas, y sus procesadores pueden ser **empotrados** (robots pequeños) o de **PC** (medianos-grandes).

Estructura de un SO: App // Middleware // Drivers // Hardware.

1.2 ARQUITECTURA DE UN ROBOT

Arquitectura de un robot: Organización de sus capacidades sensoriales, de procesamiento y de activación para conseguir un repertorio de comportamientos inteligentes interactuando con un entorno (determina el comportamiento observable del robot y se plasma en el software), tienen mejor arquitectura aquellos que poseen más comportamientos inteligentes, algunos poseen comportamientos deliberativos, pero si se tiene un comportamiento sencillo vale cualquier arquitectura.

Tipos de arquitectura: **Híbrida** (varios niveles, cambian capacidades perceptivas por deliberativas), **reactiva** (ligada a la percepción sensorial).

Simuladores: Interesa que sean realistas (todas las partes del robot son simuladas) aunque no sirvan para depurar nuestros algoritmos.

1.3 SENSORES

Sensores: Constituyen el sistema de percepción del robot, no proporcionan directamente el estado, miden cantidades físicas de varias formas además de ser limitados, ruidosos e inexactos, proporcionan símbolos en vez de señales y suele hacer falta mucha capacidad para convertir las señales a símbolos.

Características: **Sensibilidad** (ratio entre el cambio de salida y la entrada), **linealidad** (medida de la constancia del ratio/entrada/salida), **rango** (diferencia entre el máximo y el mínimo valor medible), tiempo de respuesta, exactitud, resolución, repetibilidad (fatiga), tipo de salida.

Clasificación: **Internos** (información del propio robot, posición, velocidad y aceleración), **externos** (información sobre lo que rodea al robot, proximidad, tacto, fuerza y visión), sencillos / complejos, activos / pasivos, visuales / no visuales.

1.4 CÁMARAS

Cámaras: Ofrecen un flujo desbordante de datos (sensor de luz, CCD, CMOS, blanco y negro, color, IR, imagen, resolución, etc).

Modelo Pin-Hole: Se obtiene una imagen doblemente invertida, **parámetros intrínsecos** (distancia focal, centro óptico y skew / distorsión), **parámetros extrínsecos** (prohibición y orientación, permiten sacar información espacial).

Visión en robótica: Proporciona información de medidas, potencialmente muy rico y barato, complejidad para extraer información útil y posee un flujo desbordante de datos.

Cámara RGBD: Profundidad, distancia 3D, nube de puntos, mala en exteriores debido a infrarrojos (< 6m).

LIDAR: Mide distancias 3D emitiendo pulsos láser y midiendo el tiempo de vuelo, muy raros (> 15m).

1.5 ACTUADORES

Grados de libertad: Cada uno de los movimientos independientes que realiza el robot (número de articulaciones).

Características: Generan movimiento de los elementos de un robot, la mayoría solo tienen 1 grado de libertad, **brazo humano** (más actuadores que grados de libertad), **coche** (menos actuadores que grados de libertad).

Actuadores vs Efectores: Punto final del brazo robótico.

Tipos de actuadores: Hidráulicos (más precisos), neumáticos (menos precisos), eléctricos, potencia, velocidad, precisión y controlabilidad.

} 1.6 OTROS ELEMENTOS Y MÓDULOS

Modulación PWM (anchura de pulsos): Velocidad proporcional al promedio.

Servo-motores: Capaces de colocarse en una posición, control por modulación PCM (codificación de pulsos, anchura del pulso proporcional a la amplitud deseada).

Activadores controlados: Menos sensibles a perturbaciones y proporcionan estabilidad (controladores PID).

Transmisión y reducción: Engranajes (ruedas dentadas, cremalleras y piñones, escuadras), lo que se gana en fuerza se pierde en velocidad.

} 1.7 LOCOMOCIÓN Y NAVEGACIÓN

Locomoción: Diferentes actuadores (piernas, ruedas, brazos y aletas), la locomoción con piernas es un problema muy grande para la robótica ya que el robot tiene que ser estable estática y dinámicamente.

Tipos de locomoción: Equilibrio estático y dinámico, robot holonómico (autogiro), locomoción con ruedas (mecanismo diferencial tipo tanque o ruedas directrices tipo coche).

Navegación: Manipuladores, cinemática directa (geometría), cinemática inversa (valores de las articulaciones para llegar a un punto concreto), cuellos mecánicos, PTZ.

TEMAS 2: SISTEMAS REACTIVOS Y DE CONTROL

2.1 COMPORTAMIENTO DE UN ROBOT

¿Cómo programamos el comportamiento de un robot?: Crear una máquina autónoma que cumpla un objetivo, sea eficiente y sensible al entorno. No se tienen robots para tareas domésticas ya que éstas son complejas y heterogéneas y les falta flexibilidad. Al haber uno o varios modelos se deben seleccionar bien las acciones y elegir los datos más interesantes.

2.2 SISTEMAS REACTIVOS

Características: Interacción continua con el entorno, bucle cerrado sensores-actuadores, **percepción simbólica** (árbol de posibles activaciones), **percepción subsimbólica** (sencillez, agilidad y livianidad), sin capacidad de predicción, menor capacidad de cómputo, mundo dividido en situaciones mutuamente excluyentes.

Problemas: No escalan, simplicidad de ejecución por complejidad de diseño, dificultad para superar las situaciones del mundo, uso de simplificaciones, vacíos y solapes de control, problema del arbitraje (no se consideran todas las situaciones).

Control basado en casos: Reacciona ante imprevistos, sin horizonte temporal, ágil con iteraciones rápidas, giro reactivo más robusto sin funciones largas.

Limitaciones del control: Necesita un modelo para ser preciso, no escala a comportamientos complejos o con muchos actuadores, percepción limitada.

Sistema de Reglas Borroso: Reglas lingüísticas, etiquetas lingüísticas sobre variables borrosas, valores de verdad graduales y fáciles de desarrollar.

Aprendizaje neuronal y profundo: Modelo de cómputo que puede ajustarse con datos como si fuera una red de neuronas conectadas según ciertas topologías y con unos pesos.

Redes neuronales extremo-extremo: Incluye percepción, decisiones, modelos y datos (coche autónomo sigue-línea).

Sistemas deliberativos: Actúan mediante el modelado del entorno generando un comportamiento inteligente (sistemas reactivos sin memoria).

2.3 AUTÓMATAS DE ESTADO FINITO

Autómatas de estado finito: Inteligencia dividida en estados (cada comportamiento con su percepción y condiciones), no pierden ejecución reactiva, tienen memoria, estímulos disparadores, árbol genérico.

2.4 TEORÍA DE CONTROL CLÁSICA

Teoría de control clásica: Controlar un sistema es mejorar su comportamiento para que mejore de una forma determinada, **planta** (sistema a controlar), **controlador** (modifica entradas al sistema), se basa en teorías matemáticas complejas y modelos de sistemas dinámicos en un campo complejo.

Control en lazo abierto: Actuación decidida a priori, controlador tiene un modelo matemático de la planta, no suele usar sensores, feedforward (sin vuelta atrás).

Control en lazo cerrado: Realimentación / feedback (compara continuamente el estado deseado y el actual, cuya diferencia es el error), tiene como objetivo minimizar ese error, oscila alrededor de la solución.

2.5 TIPOS DE CONTROLADORES

Controlador P ($u = -K_p \cdot e + U_b$): Tiene como objetivo anular el error, responde en proporción al error ($e = X - X_{meta}$), la ganancia se determina mediante fase experimental (si es muy alta el sistema oscila).

Controlador I ($u = K_i \cdot \int e(t) dt$): El sistema observa errores, los integra en el tiempo y cuando alcanza un umbral los corrige eliminando offsets (acumula el error con el tiempo).

Controlador D ($u = K_d \cdot de/dt$): Cuando el error disminuye se debe controlar de otra forma que cuando está aumentando.

Controlador PID: Ajustando bien las 3 constantes se obtiene una respuesta prácticamente perfecta (no se necesitan modelos de la planta).

TEMAS 3: NAVEGACIÓN DE ROBOTS MÓVILES

3.1 NAVEGACIÓN

Navegación: Ciencia de conducir un robot móvil mientras atraviesa un entorno para alcanzar un destino o meta sin chocar con ningún obstáculo.

Características: Creación / modificación de mapas, planificación de caminos, conducción (percepción del entorno, fusión de sensores, control en movimiento, esquivar obstáculos), los vehículos no holonómicos imponen restricciones adicionales.

3.2 NAVEGACIÓN LOCAL

Navegación local: Consiste en navegar sin utilizar un mapa en una dirección determinada siguiendo a alguien.

Problemas: No colisionar, seguir la dirección de otro robot, basado en sensores, destino cercano, vivaces y livianos computacionalmente, reaccionan bien ante imprevistos y no actúan hasta que el obstáculo se encuentra en su rango de visión.

3.3 TÉCNICAS DE NAVEGACIÓN LOCAL

Método de Velocidad y Curvatura (CVM): Los obstáculos se aproximan como círculos, se supone que el robot se mueve en trayectorias circulares donde el centro de la trayectoria está en el eje x, tiene como objetivo encontrar la mejor trayectoria, dirección objetiva con una velocidad rápida pero segura, la distancia entre robot y obstáculo es la tangente más corta, ventana dinámica (velocidades ideales en cada iteración).

Parámetros ($f(tv,rv)=\alpha_1*dist(tv,rv)+\alpha_2*head(rv)+\alpha_3*speed(tv)$): **dist(tv,rv)** (prefiere trayectorias que recorren más distancia), **head(rv)** (prefiere pocos cambios en la dirección objetivo), **speed(tv)** (prefiere pocos cambios en velocidad).

Método de Carriles y Velocidad (LVM): $f_s(k)=\beta_1*d(k)+\beta_2*w(k)-\beta_3*adv_{cd}(k)-\beta_4*adv_{gd}(k)$, donde **d(k)** es la distancia sin obstáculos, **w(k)** es la anchura del carril, **adv_{cd}(k)** es la preferencia por cambios pequeños en la dirección actual y **adv_{gd}(k)** es la preferencia por comandos que mantienen al robot cerca de la dirección objetivo.

Campo de potencial VFF: Fuerzas virtuales (atractivas y repulsivas), control basado en campos para pasar a vectores, utiliza mínimos locales (si las fuerzas se anulan el robot se queda atrapado).

Superposición de campos: Se calculan los campos para todo el escenario.

3.4 NAVEGACIÓN GLOBAL

Características: No es reactiva (toma tiempo para planificar la ruta), basada en mapas y no en sensores, destino lejano, localización y representación dentro y en el mapa, planificación de caminos, técnicas SLAM mediante algoritmos.

Planificación de caminos: Uso de mapas, robots móviles que se mueven en bloque de tamaño fijo con errores acumulativos.

Planificación de caminos en brazos robóticos: Espacio de configuraciones, mucha precisión, una variable por articulación.

Planes clásicos: Secuencia de subobjetivos, frágiles, no se adaptan bien a contingencias y oportunidades que surjan.

Planes como recurso: Ayuda a la decisión, es más robusto y se adapta mejor a contingencias.

Navegación basada en mapas: Dependen del formato del mapa (imagen, rejilla, etc).

} 3.5 TÉCNICAS DE NAVEGACIÓN GLOBAL

Grafos de visibilidad: Une todos los vértices entre sí además del punto inicial y final y nos quedamos únicamente con los que no tienen colisión, los obstáculos aumentan el radio del robot y su margen de seguridad, se elige el mejor camino mediante el algoritmo de Dijkstra, las aristas de los polígonos engordados también son posibles caminos, no sirven para vehículos no holonómicos.

Algoritmo: Construye un grafo de visibilidad, busca un camino entre Qinicio y Qmeta, “if solución then camino else fallo”.

Diagramas de Voronoi: Equidista de los obstáculos más cercanos maximizando la distancia con los obstáculos, se obtienen trayectorias de máxima seguridad.

Descomposición en celdas (trapezoides): Descomposición en polígonos convexos, calcular el grafo de conectividad mediante nodos (cada celda) y arcos (celdas conectadas), calcula el camino en el grafo, hace que el camino pase por el centro del lado que comparten las celdas representadas por cada nodo.

Rapidly Exploring Random Tree (RRT): Puntos aleatorios sobre el mapa y se construye el grafo a partir de ellos.

Descomposición en celdas (rejilla): Planificación por descenso de gradiente (se genera el campo en la posición destino, se propaga sin atravesar obstáculos hasta llegar al origen, se navega siguiendo el gradiente, se llega al destino descendiendo el gradiente, las celdillas cercanas a obstáculos aumentan su valor para que no pasen por ahí).

} 3.6 NAVEGACIÓN HÍBRIDA Y DE ESTÍMULO

Navegación híbrida: Establece la secuencia de subobjetivos de la navegación global y esa información se envía a la local.

Navegación siguiendo estímulo: Percepción del estímulo y de los controles reactivos, filoguiados en el entorno industrial y son capaces de seguir a una persona.

TEMA 4: MAPAS

4.1 MAPAS

Mapas: Se utilizan para representar el entorno (ocupación, calor, etc) y pueden construirse (estructura, obstáculos estáticos, largo alcance).

Representación interna del entorno: Sobre ella se toman decisiones autónomas de movimiento, sensores (información instantánea y dinámica, obstáculos, entorno local).

Uso directo de los sensores no válido: Tienen ruido e incertidumbre, dan información parcial del estímulo relevante y tienen alcance limitado y sin memoria; pero representándolos internamente si pueden cumplir estos requisitos.

4.2 TIPOS DE MAPAS

Mapas globales: Representan toda el área de movimiento del robot.

Mapas locales: Representan toda el área de visión del robot (cercanías).

Mapas topológicos: Tienen nodos y arcos (grafo de visibilidad) basados en el Algoritmo de Dijkstra.

Mapas métricos: Cuando puedes tener información en coordenadas, ángulos, distancias, etc.

Mapas de elementos geométricos: Primitivos de percepción (esquinas, segmentos, etc), las observaciones sensoriales se relacionan con las primitivas, la posición de los elementos se estima continuamente.

Mapas de rejilla: No hay primitivas solo celdillas, las observaciones sensoriales se relacionan con las celdillas, el estado de cada celdilla se estima continuamente.

4.3 TÉCNICAS DE CONSTRUCCIÓN DE MAPAS

Construcción automática de mapas: Se construyen desde los sensores y aumentan la autonomía del robot, **modelo sensorial** (cómo se interpreta la información proporcionada por el sensor), **regla de actuación** (cómo se integra la información sensorial actual con la anterior), **importancia de la localización** (integra información sensorial en el mapa, se consigue con sensores explícitos de posición o interfiriendo con sensores de rotación), **fase de exploración y fase de explotación** (separadas o imbricadas).

Mapas de rejilla (grids de ocupación): Particionan el espacio en un mallado regular de rejillas, la ocupación de cada celdilla se estima desde las observaciones sensoriales, facilita el uso de varios sensores de distinta naturaleza, permite compensar medidas ruidosas, necesita una localización buena (error < tamaño celdilla).

Geometría modelo s3nar: Tiempo de vuelo y umbral de recepci3n, informaci3n de ocupaci3n y de vac3o, reflexiones especulares (incertidumbre angular y radial), geometría axial, c3nica o lobular.

4.4 APROXIMACI3N PROBABILÍSTICA

El estado de ocupaci3n depende de la observaci3n en t y las anteriores ($P(\text{ocupada}/\text{obs}(t))$ depende del **modelo del sensor**): $P(\text{ocupada}(x,y),t) = p(\text{ocupada}/\text{obs}(t), \text{data}(t-1))$.

Ratios de probabilidad: $P(\text{ocupada}) = P(\text{ocupada}) / P(\text{ocupada}) = P(\text{ocupada}) / (1 - P(\text{ocupada}))$

Actualizaci3n de la regla de Bayes: $P(\text{mapa}(C(x,y),t)) = P(\text{obs}) / P(\text{priori}) * P(\text{mapa}(C(x,y),t-1))$

Priori es una constante: $P(\text{priori}) = P(\text{ocupada}(t)) / (1 - P(\text{ocupada}(t)))$

Pobs viene dada por el modelo del sensor: $P(\text{obs}) = P(\text{ocupada}(t)) / (1 - P(\text{ocupada}(t)))$

Inicialmente suele ser $P(\text{ocupada}) = 0.5$ y $P(\text{priori}) = 1$ cuando no sabemos c3mo es el medio.

Si $P(\text{ocupada}/\text{obs}(t))$: $0.5 P(\text{obs}) = 1$ (no cambiamos creencia), $> 0.5 P(\text{obs}) > 1$ (aumentamos la creencia), $< 0.5 P(\text{obs}) < 1$ (disminuimos la creencia).

Los mapas probabilísticos tienen inercia probabilística y han de truncarse para que tengan dinamismo.

4.5 APROXIMACI3N BORROSA

Aproximaci3n borrosa: Conjunto borroso de celdillas libres y de zonas ocupadas, cada celdilla tiene un grado de pertenencia a cada uno de ellos $\mu_l(x,y)$ y $\mu_o(x,y)$ entre 0 y 1.

Enfoque histogr3mico: Almacena valores de certidumbre $cv[0,15]$, modelo sensorial $A(t) = +3, -1$, regla aditiva ($CV_{i,j}(t+1) = CV_{i,j}(t) + A(t)$), grids dinámicos por mayoría.

Segmentos borrosos: Línea primitiva, hay que anclar su definici3n en los datos sensoriales.

TEMAS 5: AUTOLOCALIZACIÓN DE ROBOTS

5.1 LOCALIZACIÓN

Localización: Basada en sensores específicos (odometría, sensores inerciales, GPS y cámara cenital), basada en balizas, y probabilística usando mapas.

5.2 ODOMETRÍA

Odometría: Se supone que las vueltas de las ruedas pueden traducirse en desplazamientos lineales sobre el suelo (encoders absolutos y relativos).

Transformación de las coordenadas: Longitud de arcos ($AS_i = A\alpha(R-w/2)$, $AS_d = A\alpha(R+w/2)$).

Si $|AS_i| < |AS_d|$ la rueda derecha se ha movido más (centro de giro a la izquierda del robot).

Problemas: **Suposición débil** (error por deslizamiento, holgura o precisión), **errores sistemáticos** (diámetros de rueda diferentes, alineamiento de las ruedas, resolución y muestreo de encoders, se eliminan con el test del cuadrado unidireccional), **errores no sistemáticos** (patinaje de las ruedas, rugosidad y objetos en el suelo).

Sensores inerciales: Usa uno o varios acelerómetros y mide cambios en todas las orientaciones con giroscopios, típico en aviones, no necesita comunicación con nada.

5.3 GPS (TIPOS Y ERRORES)

GPS diferencial: Corrige la desviación del robot en función de la desviación recibida en una posición conocida, aumenta precisión, corrige el error con respecto a estaciones base cercanas, la información se transmite al robot.

GPS en robótica: Útil en exteriores, más interesante, se usa como un sensor más y se conecta generalmente al puerto serie.

Errores del GPS: Degradación intencional DoD, reloj, órbita, modelización de la Troposfera / Ionosfera, rebotes de señal, usuario (recepción y configuración).

5.4 BALIZACIÓN

Triangulación: Cálculo de (x, y, θ) basado en el ángulo con que se ven las balizas (arco capaz).

Trilateración: Cálculo de (x, y, θ) basado en las distancias a las balizas (GPS).

SolvePnP (Perspective n Point): Calcula la desviación y la traslación entre dos sistemas (cámara y baliza).

5.5 LOCALIZACIÓN PROBABILÍSTICA

Localización probabilística: Mide la probabilidad de que el robot esté en un sitio compatible o incompatible, el conocimiento del robot se almacena como una función densidad de probabilidad, la información se acumula en celdas (probabilidad de que el robot se encuentre ahí), probabilidad combinada con Bayes, no construye información única sino que va almacenando incertidumbre.

Función de observación: Si se combina la información con un mapa, traduce lo que se ve a información de posición (más incertidumbre en lugares menos probables), **probabilidad = $e^{-\text{distancia}}$** .

Mapas de probabilidad: **Ventajas** (buena localización desconociendo posición inicial, permite representar situaciones ambiguas), **inconvenientes** (requiere discretización del espacio, alto coste computacional, no es escalable a nuevos entornos).

Filtro de partículas: **Ventajas** (buena localización desconociendo posición inicial, permite representar situaciones ambiguas, bajo coste computacional, escalable a grandes entornos), **inconvenientes** (ajuste muy fino de parámetros).

5.6 MODELOS EN LOCALIZACIÓN

Modelo de observación: Resetea los pasos y se calculan los nuevos a la luz de la nueva observación.

Modelo de movimiento: Materializa las partículas siguiendo el movimiento.

Algoritmo de la ruleta: Cada partícula tiene un sector y un ruido Gaussiano para que no pueda clonarse, y ésta se pinta en el mapa cada vez que cae en un sector (se ejecuta n veces siendo n las partículas que tenemos).

5.7 AUTOLOCALIZACIÓN 3D

Autolocalización visual en 3D (VSLAM): Bajo coste, extrae mucha información, procesamiento muy costoso, detección de bordes y texturas, reconocimiento de formas y caras.

Autolocalización con cámaras: Visión monocular, toma puntos de referencia para autolocalizarse.

Detección de puntos (monoSLAM): Calcula puntos de interés, se comparan con otros en otro instante de tiempo para calcular el desplazamiento (Filtro de Kalman), los algoritmos necesitan tener texturas en la imagen.

PTAM: División del algoritmo en dos hilos (tracking y mapping), tracking en tiempo real, mapas optimizados con Bundle Adjustment, cálculo de hasta 1000 puntos por iteración, permite relocalización.

Deep Learning: Bounding Box, aprendizaje automático (robustez y datos masivos).