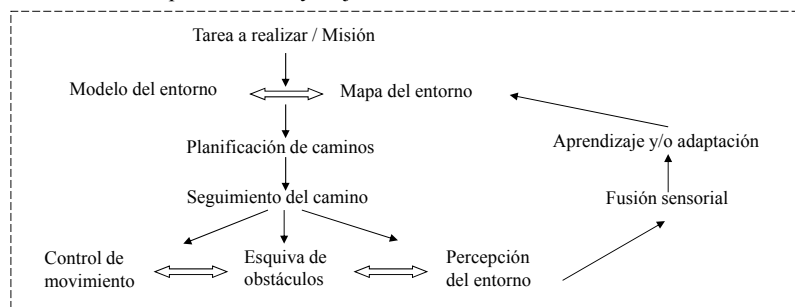


# Navegación

La ciencia (o arte) de conducir un robot móvil mientras atraviesa un entorno (tierra, mar o aire) para alcanzar un destino (o meta) sin chocar con ningún obstáculo.

Incluye tres tareas: 1.- mapeo y modelación del entorno (localización),  
2.- planificación de caminos, y  
2.- conducción

- Control: Jerarquía de tareas y flujo de información



Dpto. ACyA - Ingeniería de Sistemas y Automática

# Navegación

## Algoritmo básico

```

If there is a map then
  Search map for paths
  Select a path using an optimizing function
  If path is complex then decompose path into subgoals
  Sense environment
  While not at goal do
    Traverse path { move in direction indicated by path plan}
    If at a subgoal then obtain next subgoal
    Sense environment
    If object on path then
      Halt robot
      If object is stationary then
        Update map
        Search for alternate path to sub-goal
        If an alternate path exists then Follow alternate path
        Else Abort and replan task { no alternate path }
      Else Halt till path clears { object is moving }
    End { of while loop }
    { robot at goal }
  Else { no map – Initiate learning strategy }
    While not at goal do
      Sense environment
      If clear in direction of goal then Move toward goal
      Else { object in the way }
        If clear in other directions then
          Select a direction using a heuristic
          Move in that direction
        Else Abort and replan task { robot is trapped}
      End { of while loop }
      { robot at goal }
    End { of navigation task }
  
```

Dpto. ACyA - Ingeniería de Sistemas y Automática

## Navegación

### Maqueo, localización y guiado

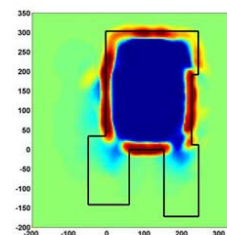
- En el problema de Navegación podemos distinguir dos situaciones:
  - Existe un mapa: podemos planificar caminos de forma global y actuar ante obstáculos imprevistos.
  - NO existe mapa:
    - Construir el mapa mientras exploramos el entorno. Debemos conocer la posición del robot o estimarla (SLAM).
    - Navegar sin mapa: control reactivo. No necesitamos la posición del robot. Actuamos ante cualquier obstáculo, mientras si no tenemos información es como si no hubiese nada.

En ambos casos debemos tener información sobre la posición del robot. Puede ser local (respecto al movimiento relativo del robot) o absoluta (respecto a un mapa existente).

## Construcción de mapas

- Nos centraremos en mapas en 2D.
- Los mapas dependen fuertemente de los sensores utilizados y de las medidas obtenidas: número de muestras, de la información que puede obtenerse, del ruido o error existente, etc.
- Se pueden distinguir tres formas de elaboración de un mapa, en función de cómo se explota la información de los sensores para elaborar el mapa del entorno:
  - mapas geométricos,
  - mapas topológicos, o jerárquicos, y
  - mapas basados en rejilla, o mapas de grid.

[http://www.iies.es/Expertos-de-la-Politecnica-de-Madrid-investigan-una-tecnica-de-construccion-de-mapas-para-robots-moviles\\_a1275.html](http://www.iies.es/Expertos-de-la-Politecnica-de-Madrid-investigan-una-tecnica-de-construccion-de-mapas-para-robots-moviles_a1275.html)

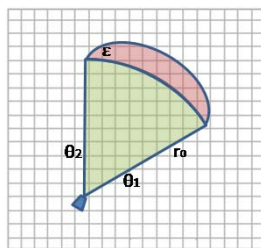


Mapa elaborado con antónimos, superpuesto con el mapa arquitectónico. El rojo indica espacio ocupado, azul espacio libre y verde espacio desconocido.

## Construcción de mapas

### Mapas basados en rejilla

- El valor de cada celda es un valor de probabilidad de ocupación o probabilidad de existencia de un obstáculo. Su actualización se lleva a cabo con la fusión de la información percibida por los sensores, y la existente en el mapa. Algoritmos de actualización: teoría bayesiana, Dempster-Shafer, filtro de Kalman, ...
- Si en el mapa se representan probabilidades de ocupación es importante disponer del modelo probabilístico del sensor. Ejemplo:

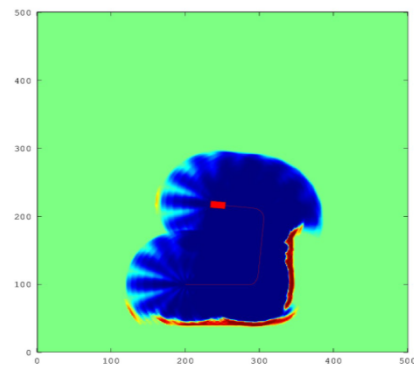
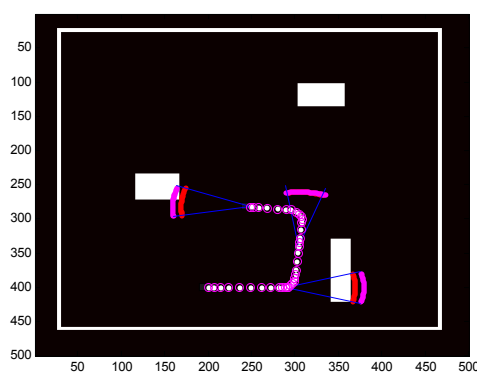


$$P(H / ocupada) = \begin{cases} 0.5 + 0.5 / k_d & \text{si } r \in [r_0, r_0 + \epsilon] \text{ y } \theta \in [\theta_1, \theta_2] \\ 0.02 & \text{si } r < r_0 \text{ y } \theta \in [\theta_1, \theta_2] \\ 0.5 & \text{Otro caso} \end{cases}$$

Dpto. ACyA - Ingeniería de Sistemas y Automática

## Construcción de mapas

### Mapas basados en rejilla. Ejemplo



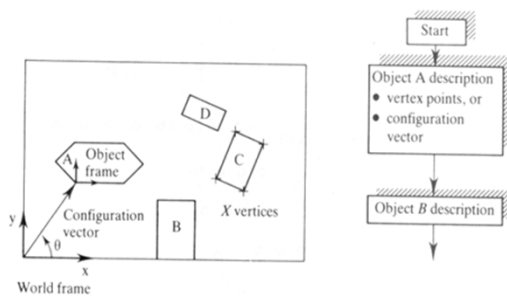
Dpto. ACyA - Ingeniería de Sistemas y Automática

## Construcción de mapas

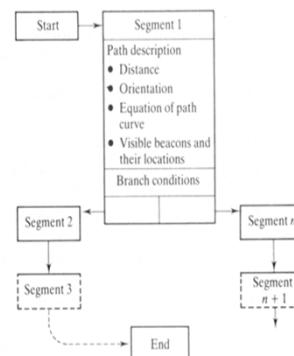
### Otros mapas. Mapas geométricos

Se utilizan cuando el entorno por el que suele moverse el robot es muy uniforme, principalmente en entornos estructurados.  
Esta representación permite que grandes extensiones puedan representarse con poca memoria.

## mapas de características



## mapas de áreas



Dpto. ACyA - Ingeniería de Sistemas y Automática

## Construcción de mapas

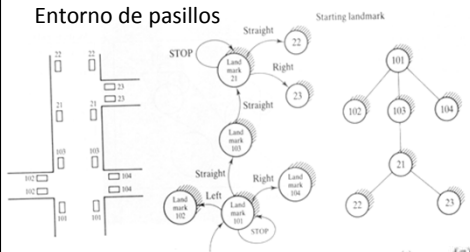
### Otros mapas. Mapas topológicos

Son básicamente un grafo ponderado:

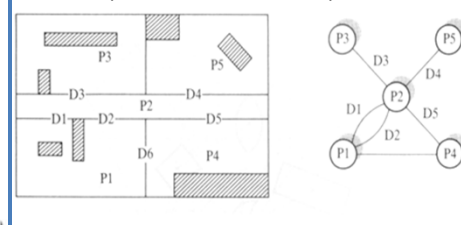
Los nodos representan situaciones diferentes del entorno.

Los arcos de conexión: indican la viabilidad de navegación entre una situación y otra contigua. El peso asignado a cada arco indica el coste de acceder de una situación a otra (ejemplo una medida de distancia).

## Entorno de pasillos



### Descripción de diferentes dependencias



### Grafo topológico a partir de un diagrama de Voronoi

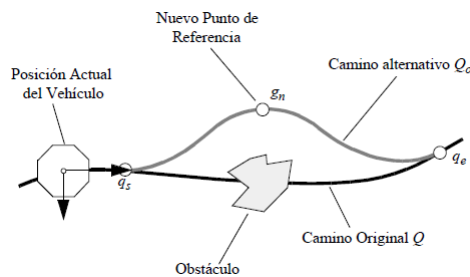


Dpto. ACyA - Ingeniería de Sistemas y Automática

## Navegación

### Esquiva de obstáculos

- Son algoritmos que se aplican en modo local. Se evitan obstáculos a medida que se detectan.



Dpto. ACyA - Ingeniería de Sistemas y Automática

## Navegación

### Esquiva de obstáculos. Ejemplos

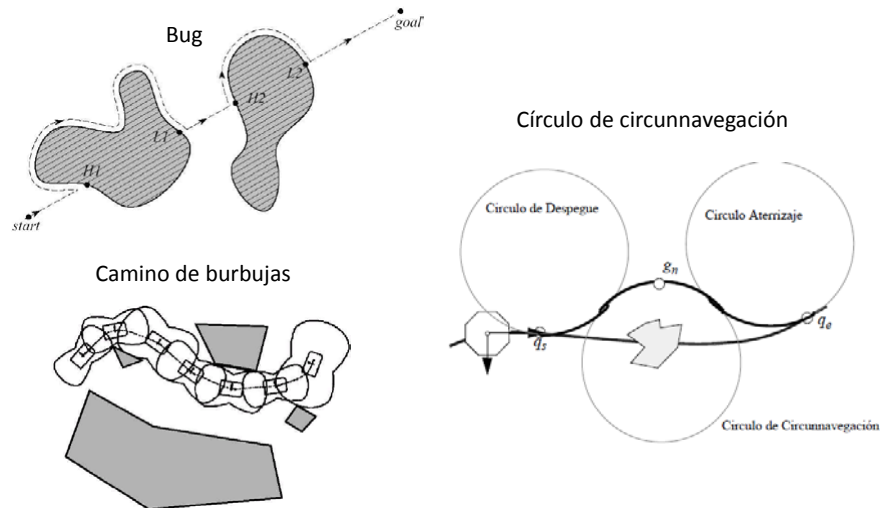
Ejemplo de métodos:

- Bug: seguir el contorno del obstáculo.
- Campo de Vectores: se crea un campo repulsivo alrededor del objeto.
- Camino alternativo: Se busca un punto de referencia que define la nueva ruta por la cual debe transcurrir el camino alternativo.
- Camino de burbujas (bubble band): se genera un “camino de burbujas” que conecta el punto inicial y el final del camino global. Burbuja es espacio que puede ser alcanzado en cualquier dirección sin colisión.
- Creación de círculo de circunnavegación: se añaden círculos de radio de giro que puede realizar el vehículo para esquivar el obstáculo.
- Etc.

Dpto. ACyA - Ingeniería de Sistemas y Automática

# Navegación

## Esquiva de obstáculos. Ejemplos



Dpto. ACyA - Ingeniería de Sistemas y Automática

## Aspectos de diseño

En el diseño y la construcción de un robot autónomo debemos tener en cuenta aspectos como:

### Software

Orden y relación de las distintas tareas que son necesarias

- Arquitectura Centralizada
- Arquitectura Descentralizada
- Arquitectura Híbrida

### ★ Hardware

Según su forma de locomoción y diseño de actuadores

- Robots de ruedas ★
- Robot con cadenas o cintas
- Robots con patas ★
- Robots submarinos y aéreos, etc.

### Comportamiento

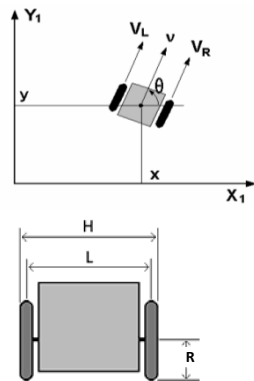
Según las tareas a realizar y su relación con otros

- Individuales
- Colectivos
- Colaborativos o cooperantes

Dpto. ACyA - Ingeniería de Sistemas y Automática

## Robots de ruedas Modelo diferencial

También conocido como robot de tipo Hilare.



Ventajas:

- El mínimo radio de giro del CIR es cero.
- Fáciles de manejar.
- Alta maniobrabilidad.
- Sencillo de implementar.

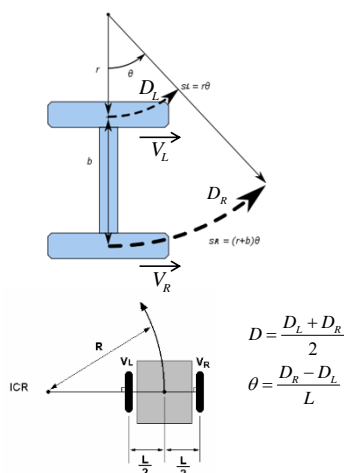
Desventajas:

- Difícil de controlar.
- No mucha precisión en rectas.



Dpto. ACyA - Ingeniería de Sistemas y Automática

## Robots de ruedas Modelo diferencial



$$D_L = \frac{2 \cdot \pi \cdot R_L}{N_C} N_L$$

$$D_R = \frac{2 \cdot \pi \cdot R_R}{N_C} N_R$$

$$V_L = R_L \cdot \omega_L$$

$$V_R = R_R \cdot \omega_R$$

Modelo cinemático y control:

Control: velocidad angular de las ruedas  $\omega_R$  y  $\omega_L$

$$v = \frac{V_L + V_R}{2}$$

$$\omega = \frac{V_R - V_L}{L}$$

$$\begin{pmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{pmatrix} = \begin{pmatrix} \cos \theta & 0 \\ \sin \theta & 0 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} v \\ \omega \end{pmatrix}$$

$$x(k) = x(k-1) + \Delta t \cdot v \cdot \cos \theta$$

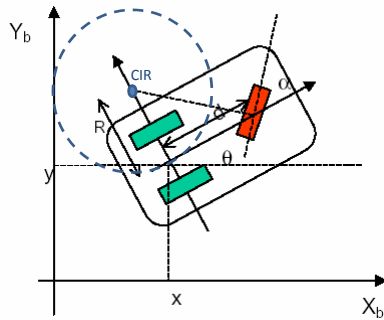
$$y(k) = y(k-1) + \Delta t \cdot v \cdot \sin \theta$$

$$\theta(k) = \theta(k-1) + \omega$$

Dpto. ACyA - Ingeniería de Sistemas y Automática

## Robots de ruedas

### Modelo triciclo



Ventajas:

- No hay deslizamiento.

Desventajas:

- El guiado es no holónimo.

Modelo cinemático y control:

Control: velocidad angular de la rueda  $\omega$  y  
Ángulo de giro de la rueda tractora  $\alpha$

$$\begin{aligned} v_{lineal} &= \omega_{rueda} \cdot R \\ v &= v_{lineal} \cdot \cos \alpha \\ w &= \frac{v_{lineal}}{d} \cdot \sin \alpha \end{aligned} \quad \begin{pmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{pmatrix} = \begin{pmatrix} \cos \theta & 0 \\ \sin \theta & 0 \\ 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} v \\ \omega \end{pmatrix}$$

$$\begin{aligned} x(k) &= x(k-1) + \Delta t \cdot v \cdot \cos \theta \\ y(k) &= y(k-1) + \Delta t \cdot v \cdot \sin \theta \\ \theta(k) &= \theta(k-1) + \omega \end{aligned}$$

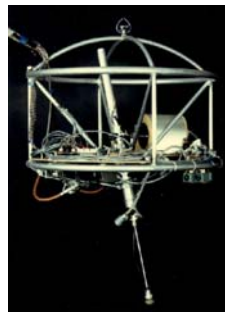
Dpto. ACyA - Ingeniería de Sistemas y Automática

## Robots de patas

Existen robots sin patas ni ruedas (que se arrastran), con una pata, dos, tres, cuatro, seis, ocho...



SILO 4



3D One-Leg Hopper



SILO 6



Dpto. ACyA - Ingeniería de Sistemas y Automática



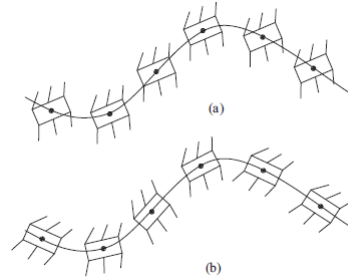
# Robots de patas

## Modos de caminar. Generalidades

Vamos a considerar garantizar estabilidad estática (tres patas siempre en el suelo):

Un robot caminante *es un móvil omnidireccional*: el **seguimiento de trayectorias** lo puede realizar de diferentes maneras:

- a) Modo de deriva: el centro del sistema de referencia del cuerpo puede seguir una trayectoria manteniendo constante la orientación de su eje longitudinal
- b) Modo giratorio: manteniendo este eje tangente a la trayectoria.



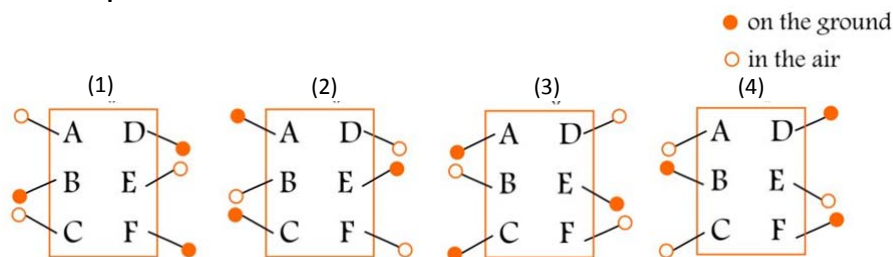
A parte de un generador de modos de caminar estable, el controlador del robot tiene que realizar otras tareas como el control de inclinación y elevación del cuerpo.

**Modos de caminar:** *periódicos* (movimiento se repite ciclicamente) o *aperiódicos* (movimiento depende de situación y terreno). Además pueden ser *continuos* (el cuerpo se mueve a la vez que las patas) o *discontinuos* (el cuerpo está parado mientras se mueven las patas).

# Robots de patas

## Modos de caminar. Ejemplo

### Modo periódico continuo:



- (1): Patas A,C,E hacia delante. Patas D,F,B hacia atrás.
- (2): Patas A,C,E se bajan. Patas D,F,B se suben.
- (3): Patas A,C,E hacia atrás. Patas D,F,B hacia delante.
- (4): Patas A,C,E se suben. Patas D,F,B se bajan.