

Navegación

Introducción

La navegación es 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

- Creación de mapas
- Planificación de caminos
- Conducción
 1. Percepción del entorno
 2. Fusión de sensores
 3. Modificación/generación de mapas
 4. Control de movimiento
 5. Esquivar obstáculos

Navegación global

- Destino lejano
- Necesidad de mapa: a priori o construido automáticamente
- Localización dentro de ese mapa
- Se apoya en una representación del entorno
- Planificación de caminos
- SLAM: Simultaneous Localization and Mapping

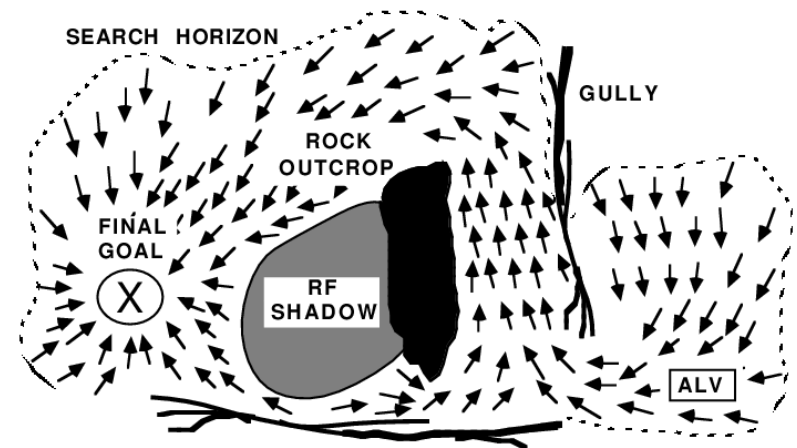
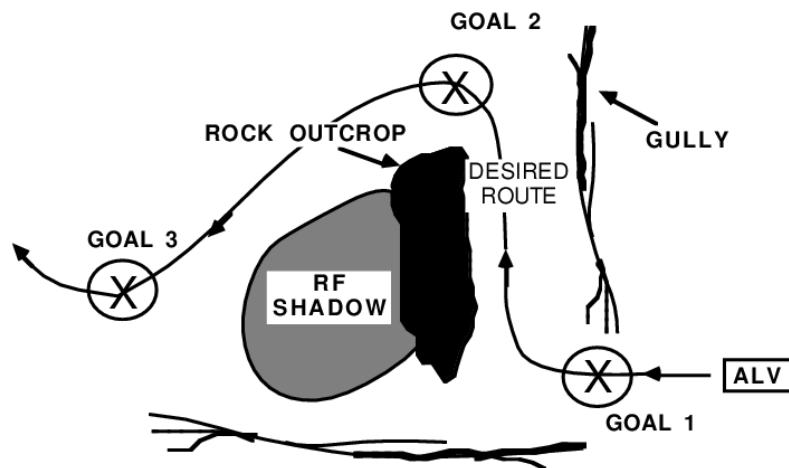
Planificación de Caminos

Manipuladores vs. robots móviles:

- Problema equivalente pero en manipuladores
- Los robots móviles se mueven en bloque
- El tamaño (salvo excepciones) es fijo en los móviles
- En manipuladores se busca mucha precisión
- En los móviles los errores son acumulativos
- Los sensores del móvil tienen campos de visión variables

Planes clásicos y planes como recurso

- clásico: secuencia de subobjetivos
- recurso: ayuda a la decisión



Navegación basada en mapas

- Dependen del formato del mapa
- Grafo de visibilidad
- Diagramas de voronoi
- Descomposición en celdas

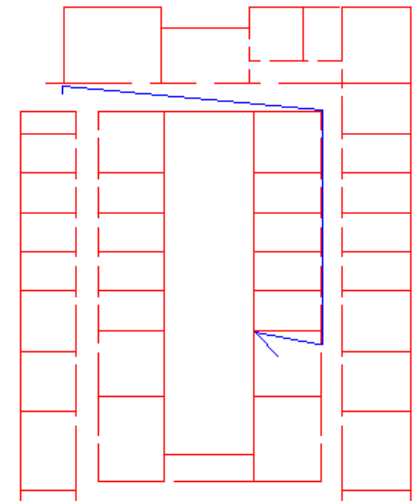
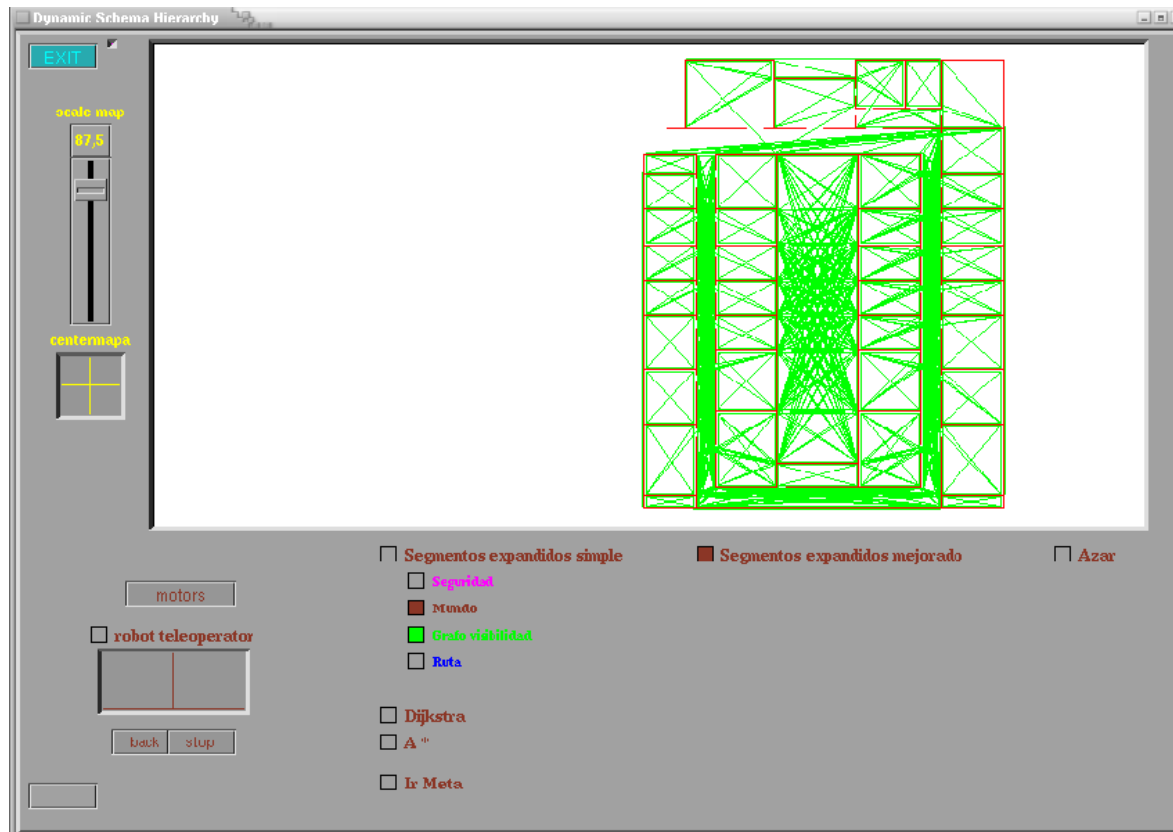
- A*
- Dijkstra
- Gradient Path Planning

Listas de Caminos, grafos de objetos o balizas

- Lista de Caminos, puede incluir información como:
 - Zonas vacías
 - Zonas con gente
 - Zonas peligrosas
- Grafos de objetos, idem pero de objetos
- También con balizas
- Permiten búsquedas muy eficientes
- Pueden existir regiones ambiguas

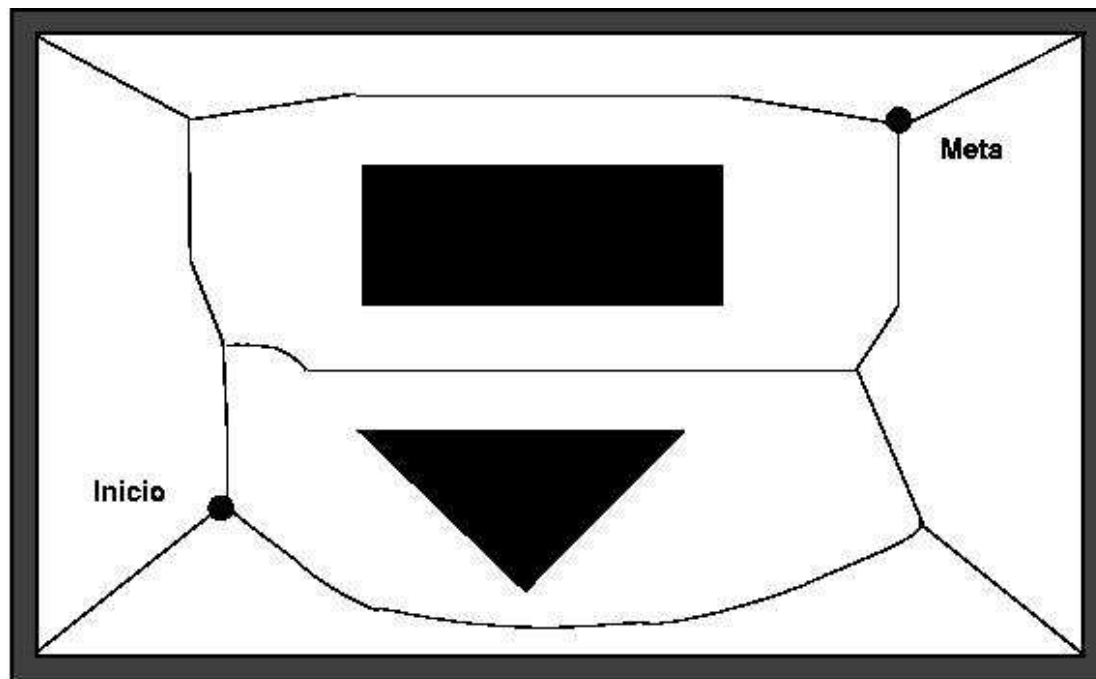
Grafos de Visibilidad (Nilsson 1969)

- Uno de los más antiguos
- Une vértices de obstáculos (polígonos) mediante rectas que no atraviesan obstáculos
- Método 2D
- Se obtienen caminos de distancia mínima
- Algoritmo:
 - 1 Construir grafo de Visibilidad G :
 - $G \longrightarrow$ grafo no dirigido construido como
 - Nodos: q_{inicio} , q_{meta} , y vértices de los polígonos
 - Arcos: uno por cada línea que no atraviesa un objeto
 - 2 Buscar camino entre q_{inicio} y q_{meta}
 - 3 **if** solución **then** Camino encontrado **else** fallo



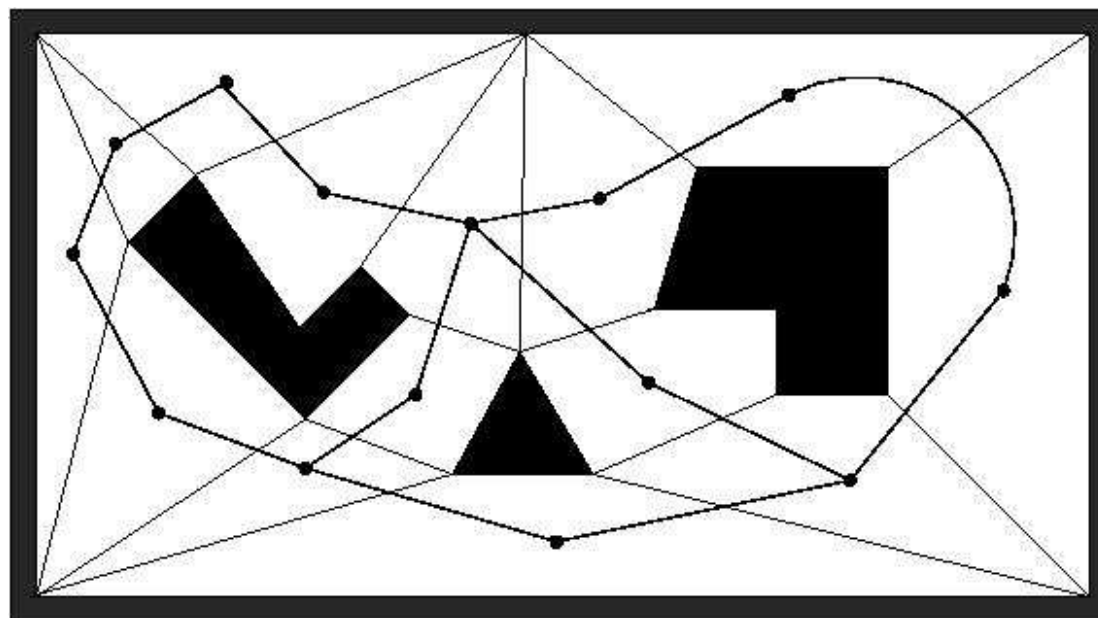
Diagramas de Voronoi

- Los puntos de cada celda están más próximos a un objeto que a los demás (paredes, obstáculos)
- El algoritmo es similar: identificar arcos entre q_{inicio} y q_{meta}
- Se obtienen trayectorias de máxima seguridad

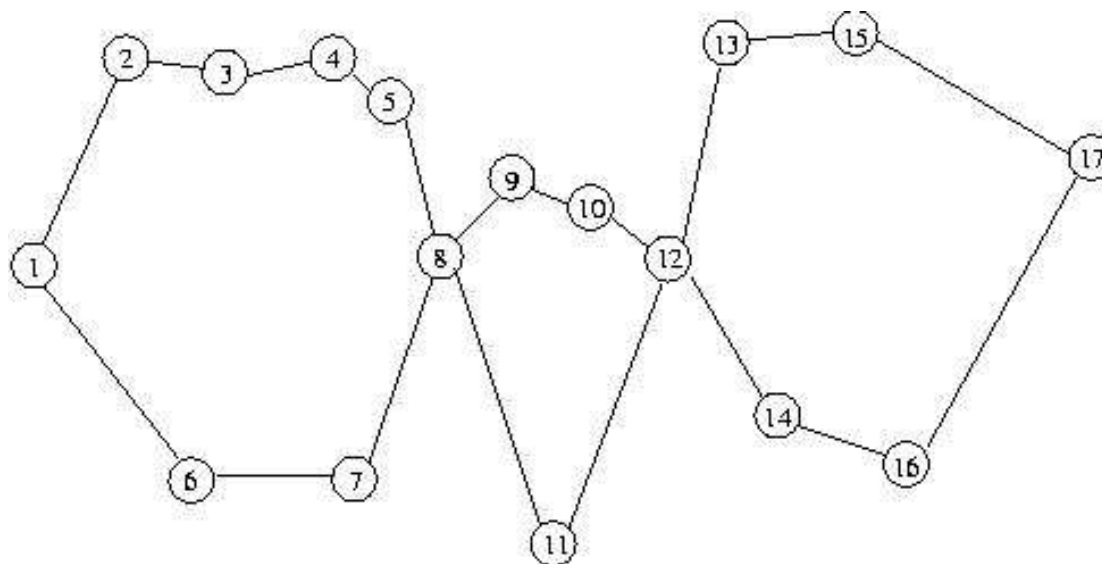


Descomposición en celdas (Trapezoidales)

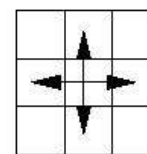
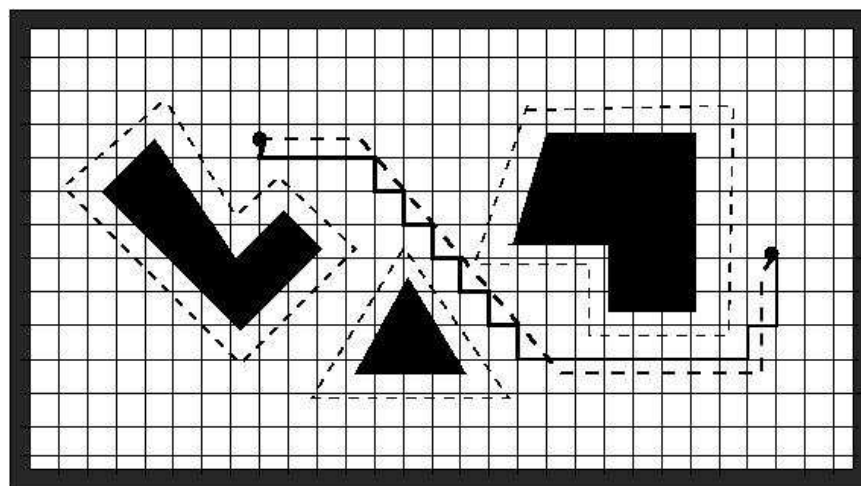
- Descomposición en polígonos convexos
- Calcular el grafo de conectividad
 - Nodos: Cada celda
 - Arcos: Si las celdas están conectadas
- Calcular el camino en el grafo
- Hacer que el camino pase por el centro del lado que comporten las celdas representadas por cada nodo



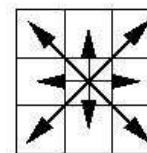
Descomposición en celdas (Verticales)



Descomposición en celdas (Enrejado)



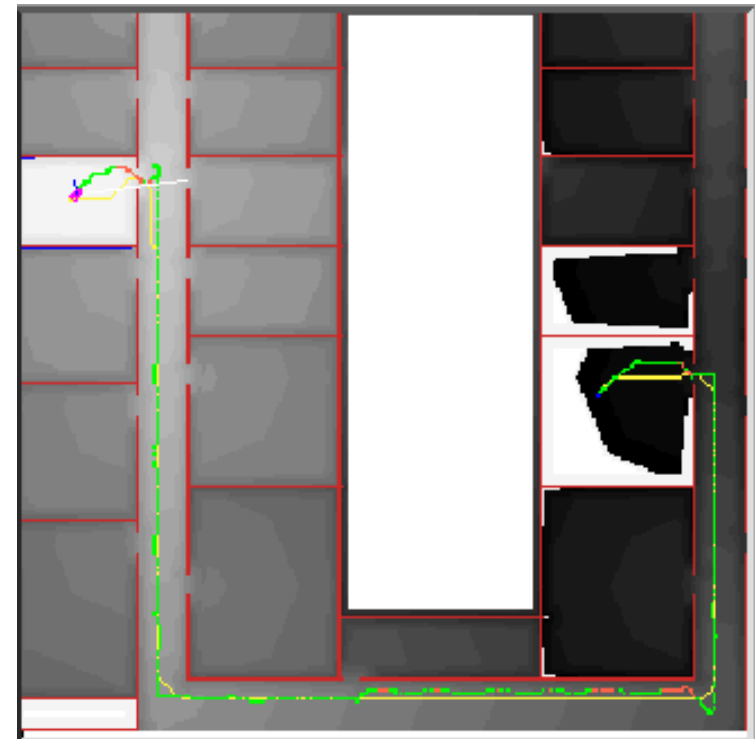
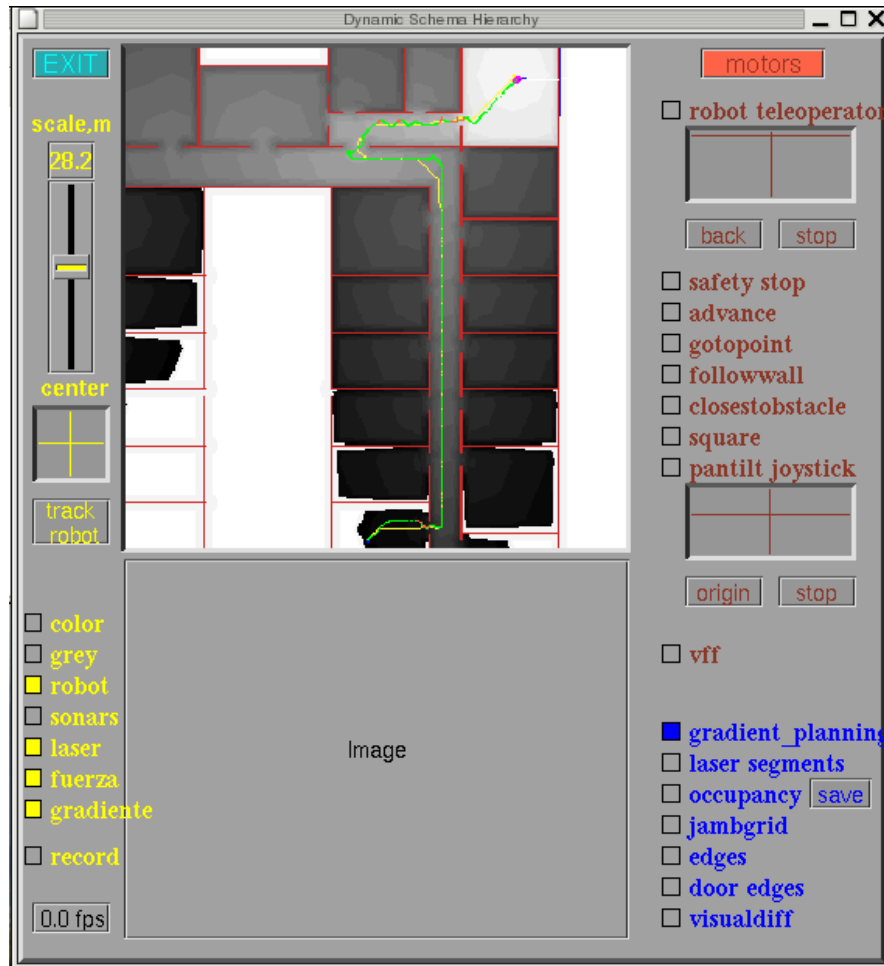
4-Connectado



8-Connectado

Planificación por descenso del gradiente

- Se genera el campo en la posición del destino
- El campo se propaga como una onda, sin atravesar los obstáculos
- Los propios obstáculos pueden subir el valor del campo en su cercanía
- La propagación se detiene cuando se llega al punto de partida
- Se navega siguiendo el gradiente de ese campo:
 - se acerca al destino
 - se mantiene alejado de los obstáculos



Navegación local

Navegar sin usar un mapa

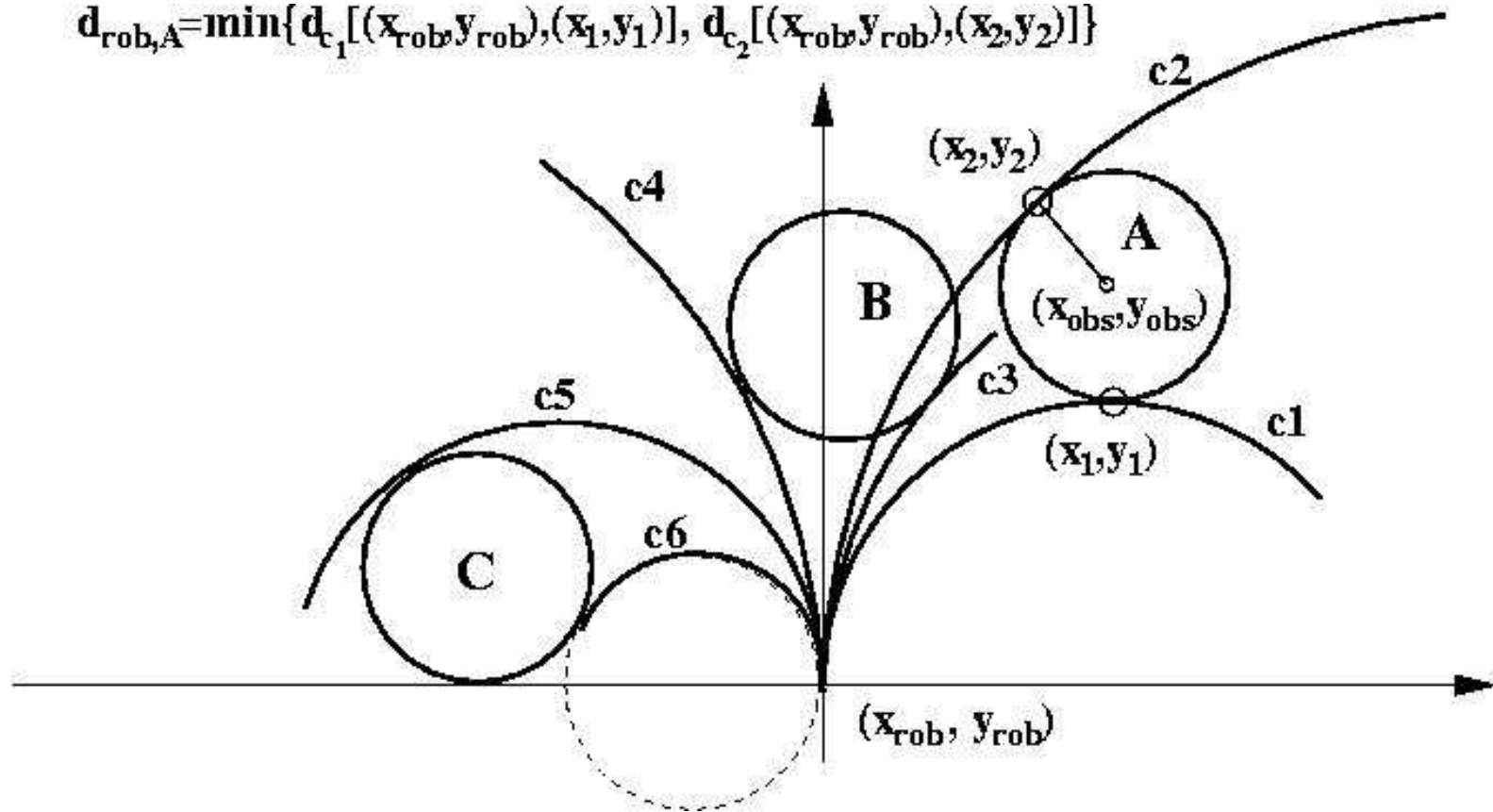
- Sin un mapa ¿dónde podemos ir?
 - En una dirección determinada
 - Siguiendo a otro
- ¿Cuáles son los problemas?
 - Generalmente no colisionar
 - Seguir la dirección, al otro robot
 - Uso por otros comportamientos

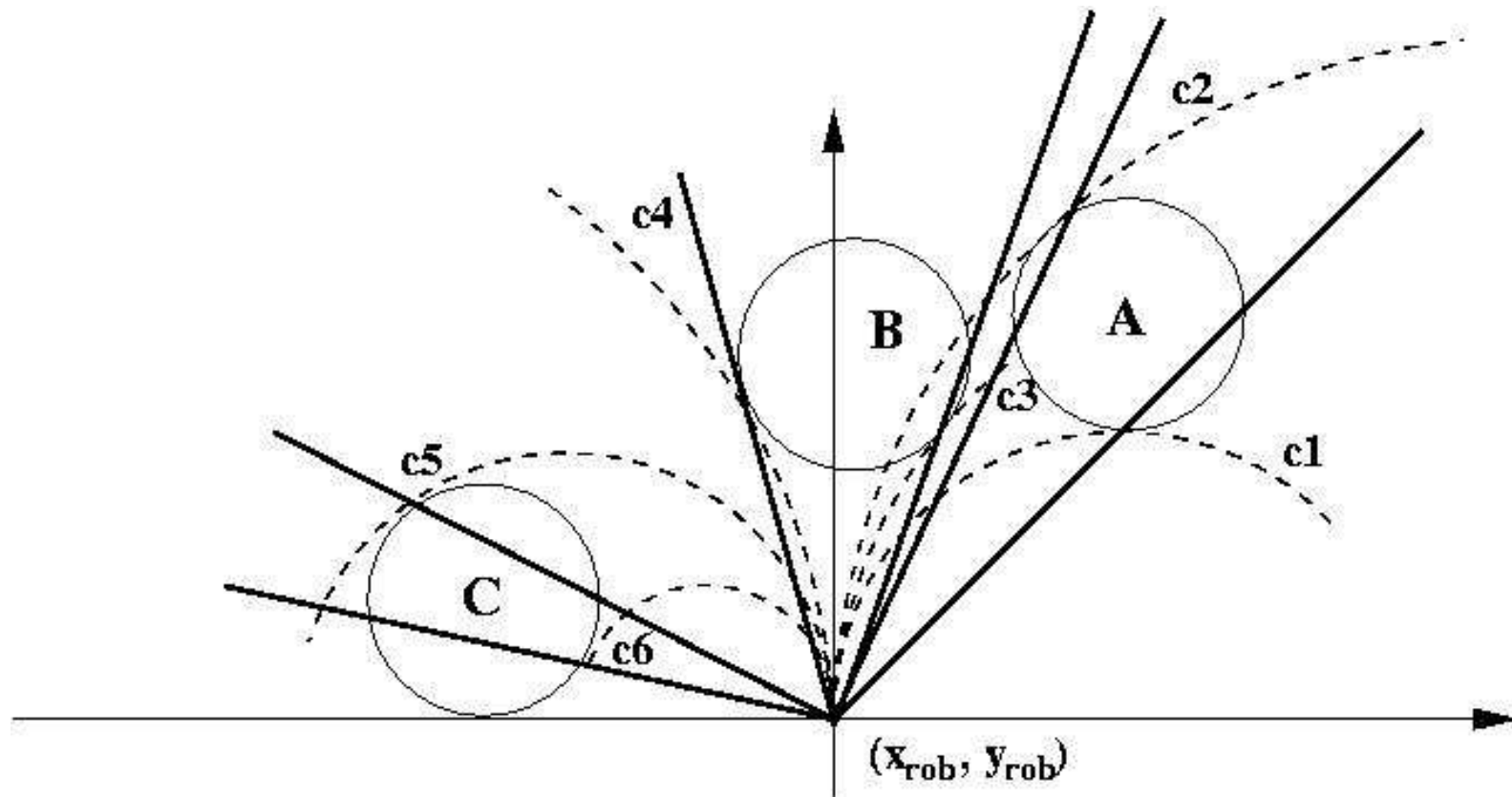
- Basada en sensores
- Destino cercano
- Vivaces, livianos computacionalmente, 10Hz
- Reaccionan bien ante imprevistos
- Miopía: hasta que no ve un obstáculo con los sensores no maniobra para esquivarlo
- CVM, espacio de velocidades
- LVM
- Campos de potencial, VFF
- Superposición de campos

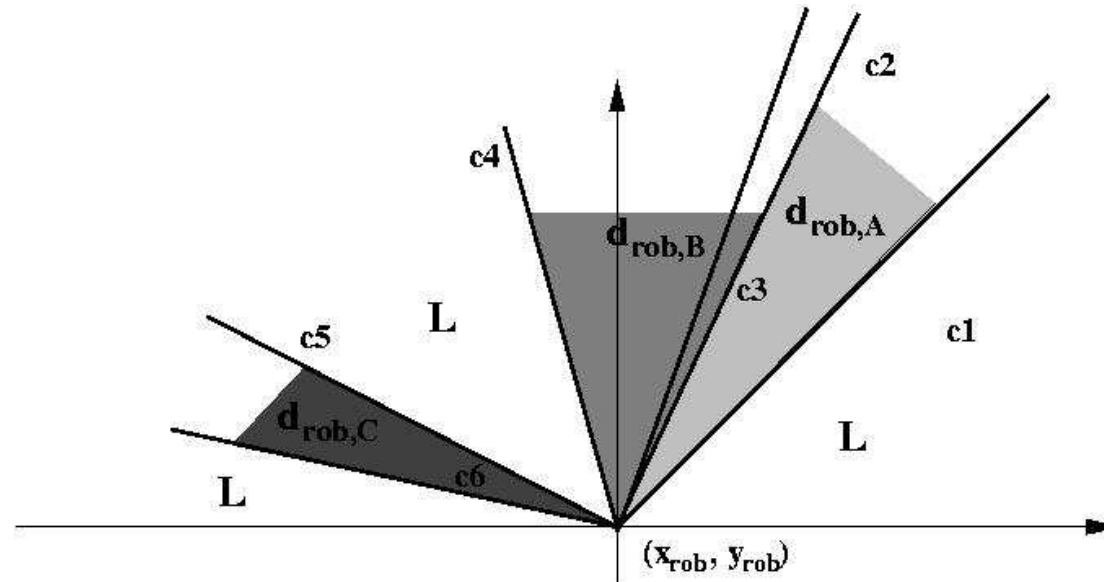
El método de Velocidad y Curvatura (CVM)

- Los obstáculos se aproximan a círculos
- Supondremos que el robot se mueve en trayectorias circulares
- El centro de esas trayectorias está en el eje X
- El objetivo será encontrar la “mejor” trayectoria
- El robot tiene una dirección objetivo
- En la figura (x_{rob}, y_{rob}) es la posición del centro del robot
- La distancia del robot a un obstáculo es la tangente más corta

$$d_{rob,A} = \min\{d_{c_1}[(x_{rob}, y_{rob}), (x_1, y_1)], d_{c_2}[(x_{rob}, y_{rob}), (x_2, y_2)]\}$$







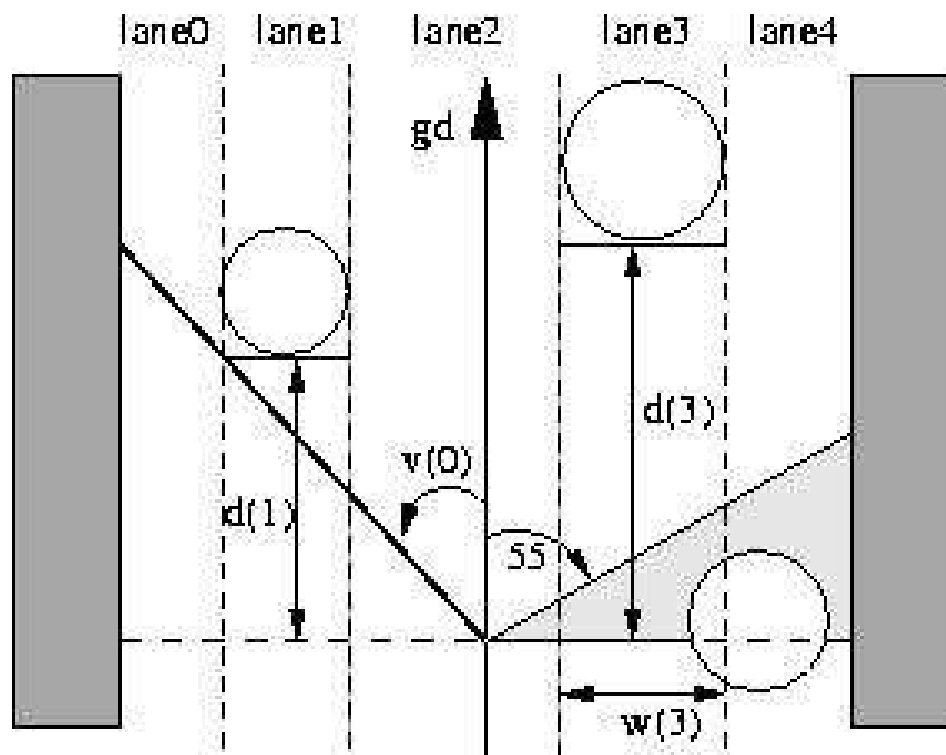
$$f(tv, rv) = \alpha_1 dist(tv, rv) + \alpha_2 head(rv) + \alpha_3 speed(tv)$$

Donde:

- $dist(tv, rv) = D_{limit}(tv, rv, OBS)/L$
- $head(rv) = 1 - |\theta_c - rvT_c|\pi$
- $speed(tv) = tv/tv_{max}$

- Los sectores nos sirven para ver obstáculos superpuestos
- Los sectores coloreados representan la distancia máxima en ese sector
- $dist(tv, rv)$ prefiere las trayectorias que recorren la mayor distancia
- $head(rv)$ prefiere los menores cambios con respecto a la dirección objetivo
- $speed(tv)$ prefiere los cambios menores de velocidad
- Los tres factores están normalizados (p.e. la distancia con respecto a L , distancia máxima).
- Los factores α_i ponderan de forma *heurística* los tres criterios

El método de Carriles y Velocidad (LVM)



$$f_s(k) = \beta_1 d(k) + \beta_2 w(k) - \beta_3 ad_{v,cd}(k) - \beta_4 ad_{v,gd}(k)$$

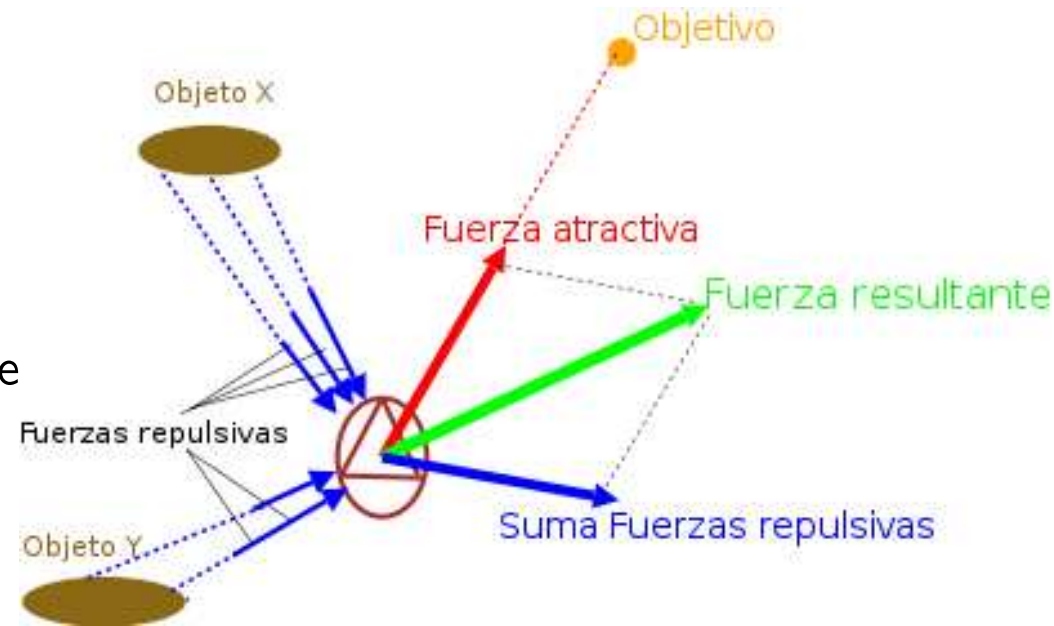
$$f_s(k) = \beta_1 d(k) + \beta_2 w(k) - \beta_3 ad_{v,cd}(k) - \beta_4 ad_{v,gd}(k)$$

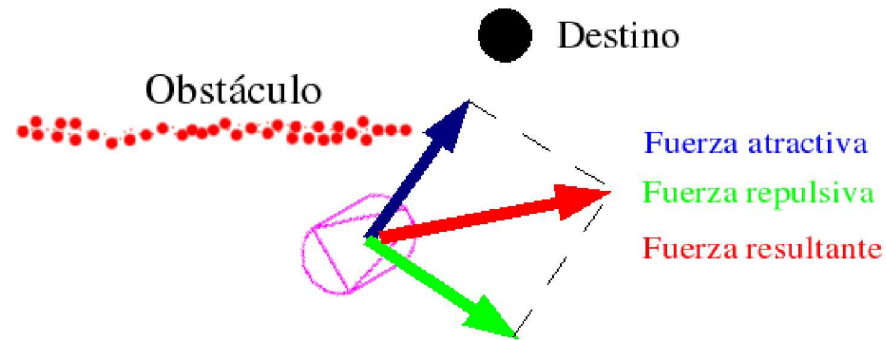
donde:

- $d(k)$ distancia sin obstáculos en el carril k
- $w(k)$ anchura del carril k
- $ad_{v,cd}(k)$ indica la preferencia por cambios pequeños en la dirección actual.
- $ad_{v,gd}(k)$ indica la preferencia por comandos que mantienen la robot cerca de la dirección objetivo (gd).

Campos de Potencial, VFF

- Fuerzas virtuales
- F. repulsivas de los obstáculos
- F. atractiva del destino
- Suma vectorial ponderada
- Se calcula en el punto donde está el robot en cada momento
- Borenstein, *Virtual Force Field*
- Campos de potencial





- Repulsión inversamente proporcional a distancia
- Atracción constante, salvo cerca que es proporcional a distancia
- Suma ponderada: timidez vs prudencia

$$F_{resultante} = \alpha F_{atractiva} + \beta F_{repulsiva}$$
- Traducir $F_{resultante}$ a órdenes para los motores (basado en casos)
- Mínimos locales
- Oscila

Superposición de campos

