Robots Móviles

Tema 4.3. Evitación de obstáculos

Otto Colomina Pardo

otto@dccia.ua.es

Índice

Evitación de obstáculos

- Necesaria ya que habrá **obstáculos que no estén en el mapa**
- Necesitamos técnicas que respondan en **tiempo real**
  - Solo podrán usar **información local**
  - Problema: soluciones **subóptimas**
  - Tendrán cierto parecido con los comportamientos de la **robótica reactiva**

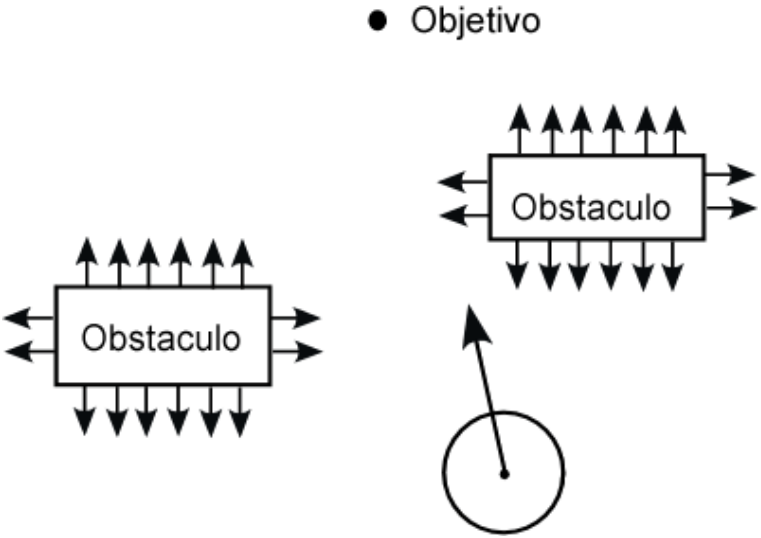
Clasificación de técnicas

- Métodos de campo de fuerzas
  - Similares a los campos de potencial, aunque algo más sofisticados
- Métodos en el espacio de velocidades
  - Tienen en cuenta la dinámica del robot. El más conocido es el de la ventana dinámica

Índice

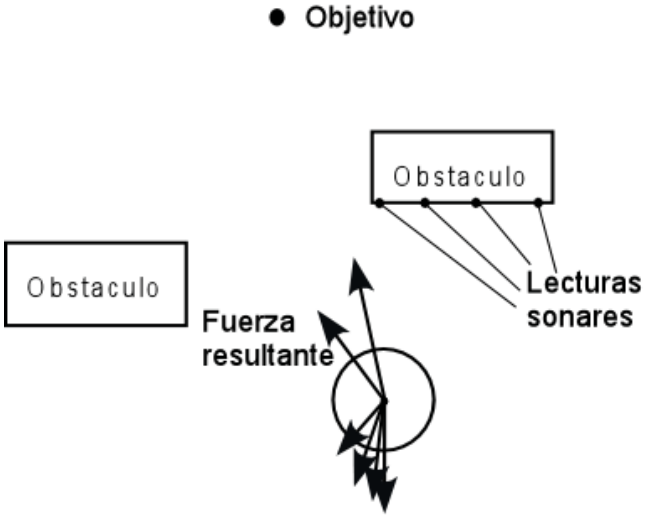
Métodos de campo de fuerza

- Misma idea básica que en robótica reactiva con campos de potencial
- Dos tipos de fuerza: atractiva y repulsiva



Fuerzas actuando conjuntamente

- La fuerza de **repulsión** es inversamente proporcional a la distancia de la lectura del sensor
- La fuerza de **atracción** es constante (salvo quizá en la vecindad del objetivo)



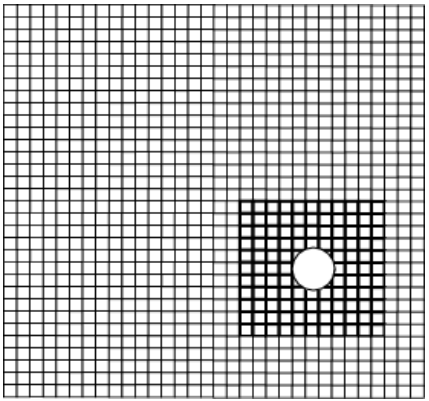
Problema: ruido en los sensores

- Una lectura de un sensor puede no ser un obstáculo real sino **ruido**

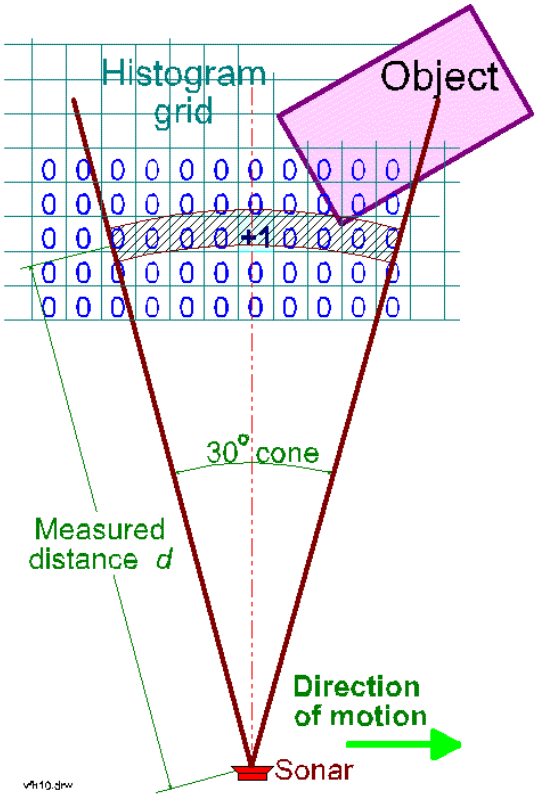
- Si hacemos que la fuerza repulsiva dependa de las lecturas instantáneas el comportamiento puede ser **inestable**

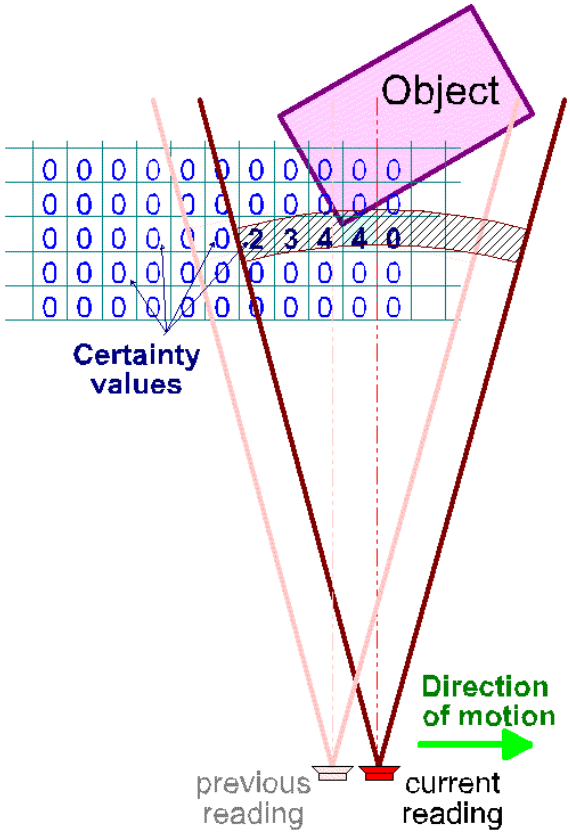
Método del VFF (Vector Field Force)

- Borenstein y Koren, 1989
- Integra el campo de potencial con una especie de **rejilla de ocupación** local alrededor del robot
  - Las celdas empiezan en 0
  - Cada vez que se detecta una lectura en la zona de una celda, se incrementa su valor en 1



VFF: actualización de la rejilla





VFF: cálculo de la fuerza resultante

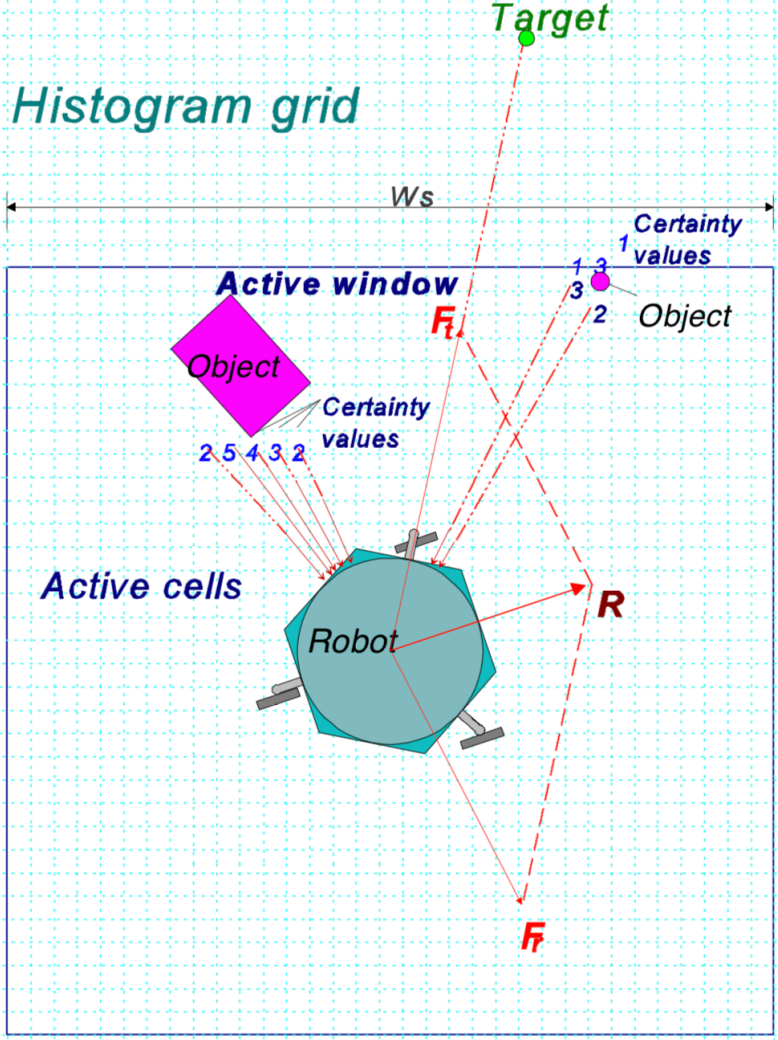
Cada celda en (i,j) contribuye con una fuerza repulsiva

$$F(i,j) = \frac{F_{cr} C(i,j)}{d^2(i,j)} \left[\frac{x_i - x_r}{d(i,j)} \hat{x} + \frac{y_i - y_r}{d(i,j)} \hat{y} \right]$$

La fuerza repulsiva total es la suma de las F de cada celda

$$F_r = \sum_{i,j} F(i,j)$$

La fuerza atractiva (F_t en la figura) es constante



VFF: comando de giro

- Calculamos δ, el ángulo del vector de la fuerza resultante con el eje X
- El comando de velocidad angular a suministrar al robot es

$$\omega = K_s[\theta(-)\delta]$$

donde

- Ks es una constante proporcional de giro
- θ es la dirección actual del robot
- (-) es la diferencia rotacional más cercana entre θ y δ en el intervalo [-180,180]

Suavizado del comando de giro

Para evitar cambios bruscos en el comando de giro se puede utilizar un filtro paso bajo

$$\omega_i = \frac{T\omega'_i + (\tau - T)\omega_{i-1}}{\tau}$$

donde

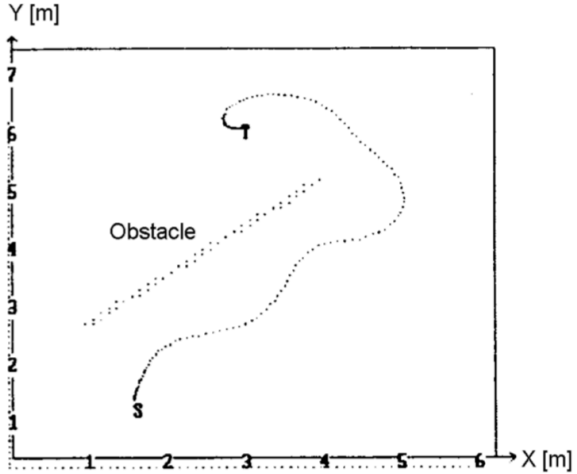
- ω_i es el comando de giro después del filtrado
- ω'_i ídem antes del filtrado

  ω_{i-1} ídem en el instante anterior

  T tiempo de muestreo
- τ constante de tiempo del filtro

Problema: movimiento oscilatorio

- Hay un “retraso” en la reacción del robot debido a inercia + filtro, lo que da lugar a **oscilaciones**



- Idea: **reducir la fuerza repulsiva** si el robot no va hacia el obstáculo

Reducir Fr

- Calcular ángulo entre vector velocidad y vector Fr

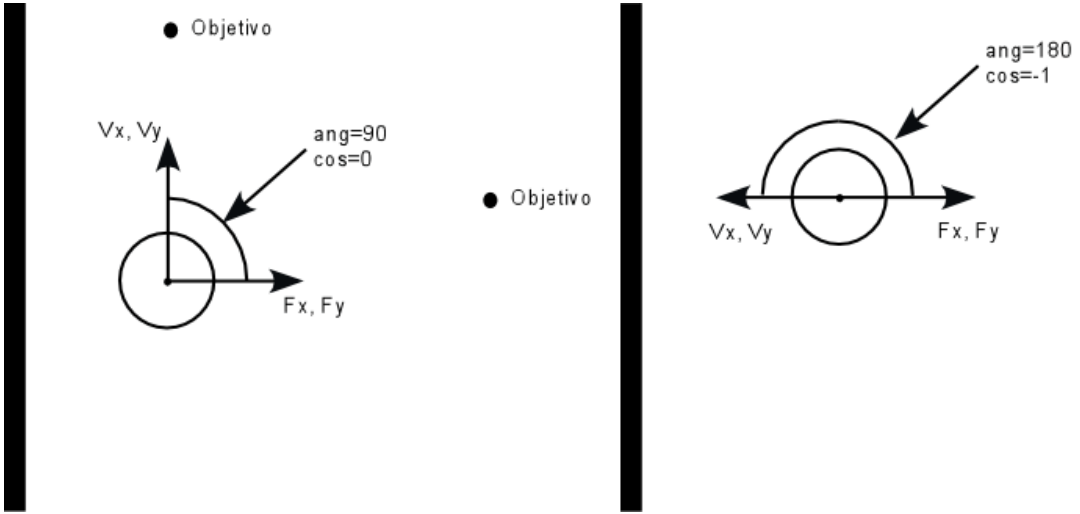
$$\cos(\theta) = \frac{V_x F_{r_x} + V_y F_{r_y}}{|V||F_r|}$$

- Calcular Fr ajustado:

$$\overrightarrow{F_r'} = w\overrightarrow{F_r} + (1 - w)\overrightarrow{F_r}(-\cos\theta)$$

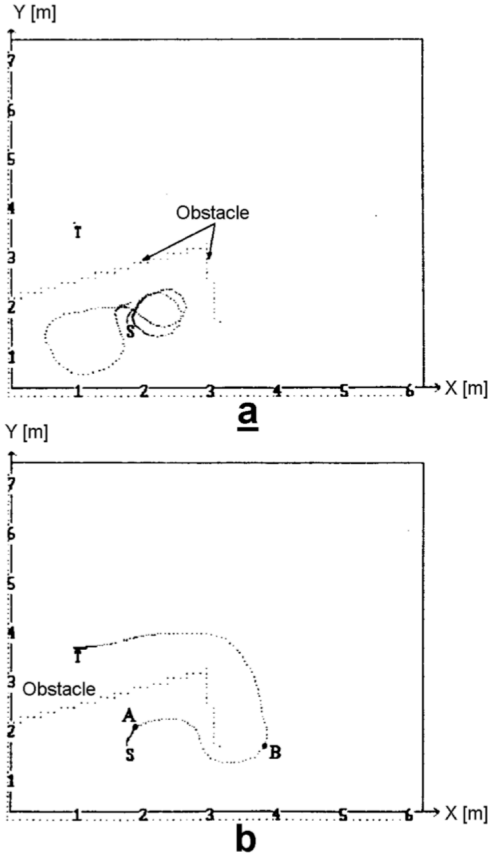
Modificación de la velocidad lineal

$$V = \begin{cases} V_{max} & \text{para } |F_r| = 0 \text{ (no hay obstáculos)} \\ V_{max}(1 - |\cos(\theta)|) & \text{para } |F_r| > 0 \end{cases}$$



Problema: mínimos locales

- Si la diferencia entre la dirección de movimiento y la dirección al destino es > 90º probablemente estamos “dando vueltas”
- En ese caso el VFF se pone en modo “seguir pared”



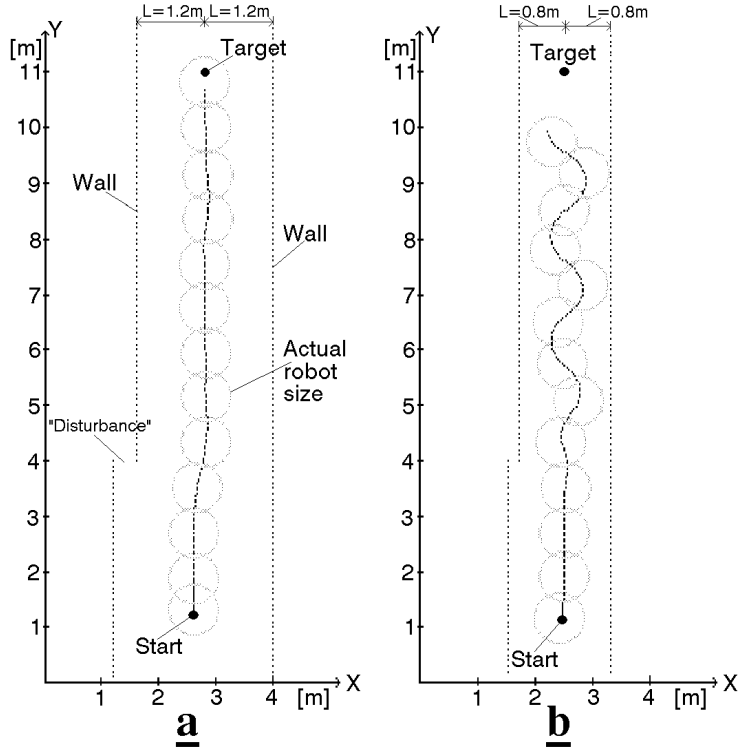
Resumen del VFF

- Se mejora la respuesta frente a ruido: no respondemos ante una lectura sino ante la acumulación de evidencia de varias de ellas

- Se introducen métodos para suavizar la respuesta del robot
  - Filtro paso bajo para giro
  - Amortiguación Fr

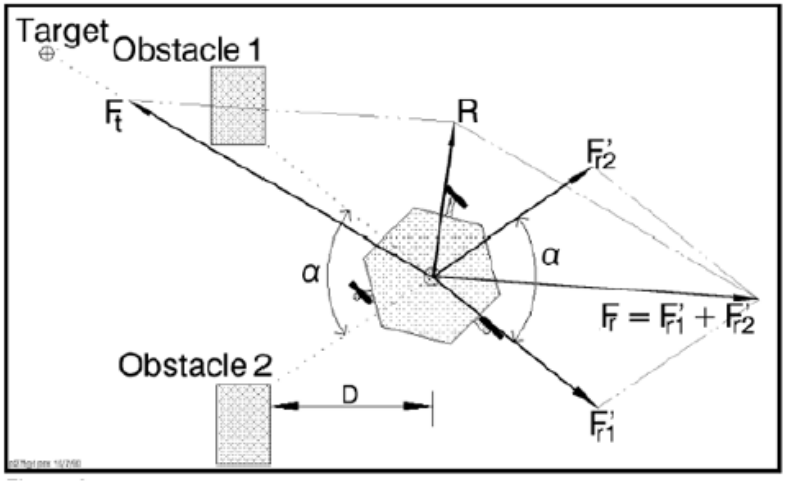
El VFF sigue teniendo problemas

- Movimiento **inestable** en ”zonas estrechas”
- Le “cuesta” **entrar por puertas** porque las paredes lo repelen



El VFF sigue teniendo problemas

- Movimiento **inestable** en ”zonas estrechas”
- Le “cuesta” **entrar por puertas** porque las paredes lo repelen



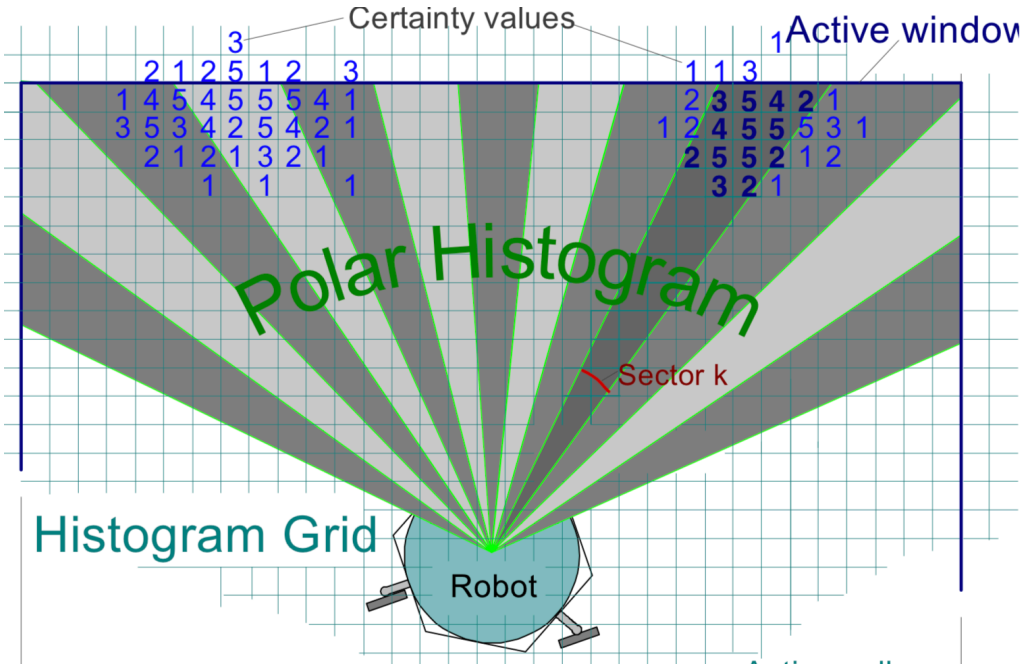
¿Por qué estos problemas?

Careful analysis of the shortcomings of the VFF method reveals its inherent problem: **excessively drastic data reduction** that occurs when the individual repulsive forces from histogram grid cells are added up to calculate the resultant force vector F_r. **Hundreds of data points are reduced in one drastic step to only two items**: direction and magnitude of F_r . Consequently, detailed information about the local obstacle distribution is lost.

Según los propios Borenstein y Koren

Mejora: VFH (Vector Field Histogram)

- Reducción de datos en *dos fases*
  - *Rejilla -> histograma polar -> F*



Fases del método

- Se actualiza el valor de las celdas, tal como lo hacíamos en el VFF
- Se transforma el espacio **2D en 1D**, construyendo un histograma polar
- Se **umbraliza** el histograma polar para observar obstáculos en determinadas posiciones

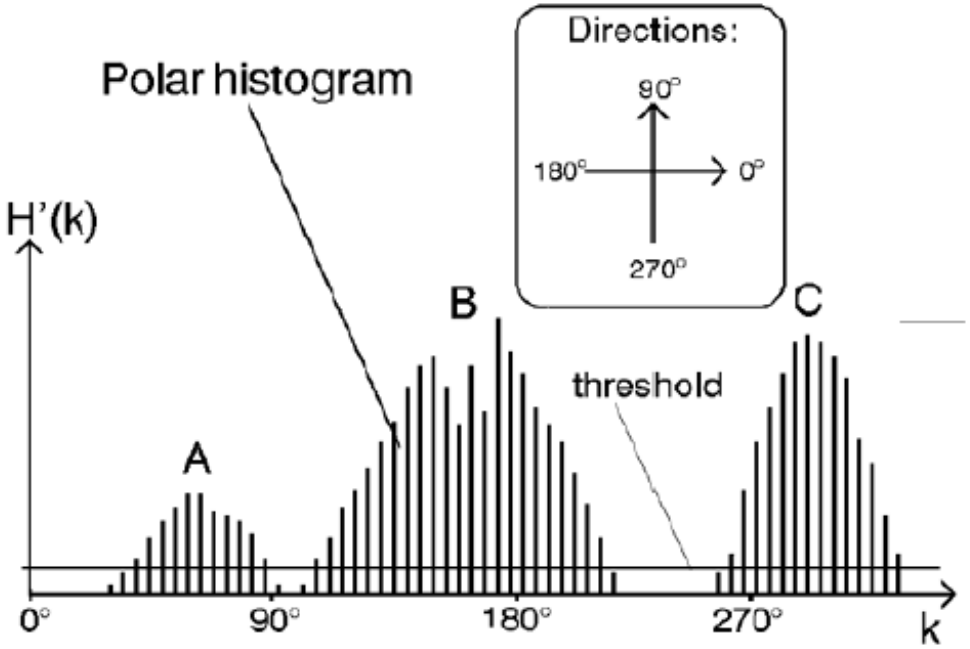
Transformación 2D a 1D

- La subrejilla alrededor del robot se divide en sectores angulares (5º)
- El vector 1D tendrá tantas componentes como sectores angulares
- Cada componente acumula evidencia en el sector angular correspondiente (histograma)

Transformación 2D a 1D

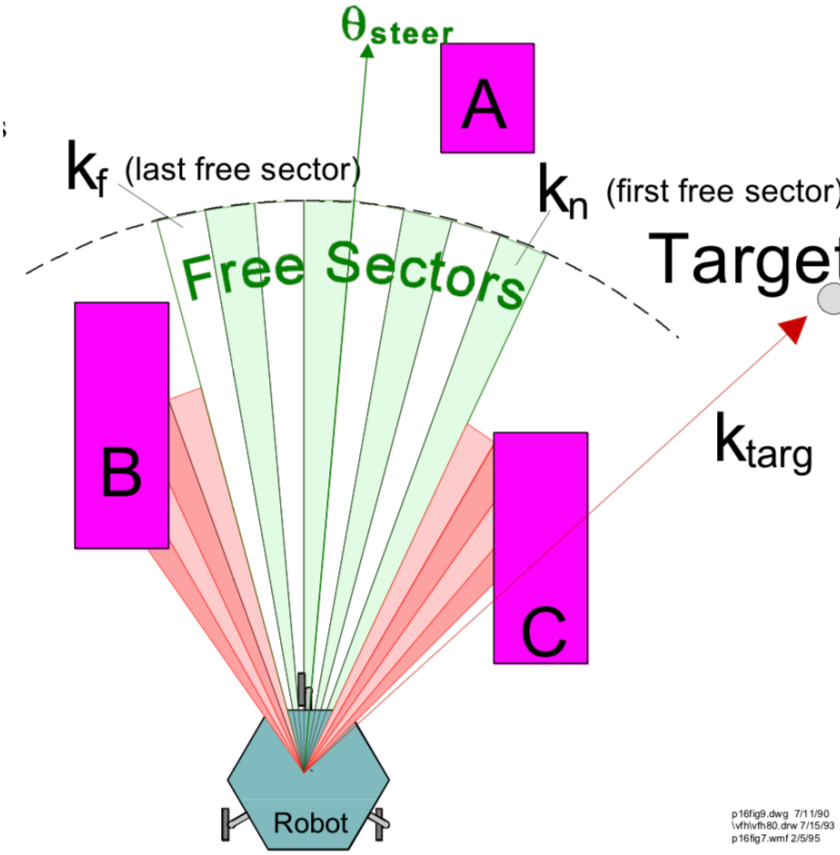
- El ángulo de una celda se calcula $\beta_{i,j} = \arctan \frac{y_j - y_r}{x_j - x_r}$
- y la magnitud de una celda $m_{i,j} = (C(i,j)(a - bd_{i,j})$
- donde
  - a, b son constantes positivas
  - $C(i,j)$ es el valor de la celda (i,j)
  - $d_{i,j}$ es la distancia de la celda al robot

Histograma calculado y umbralizado



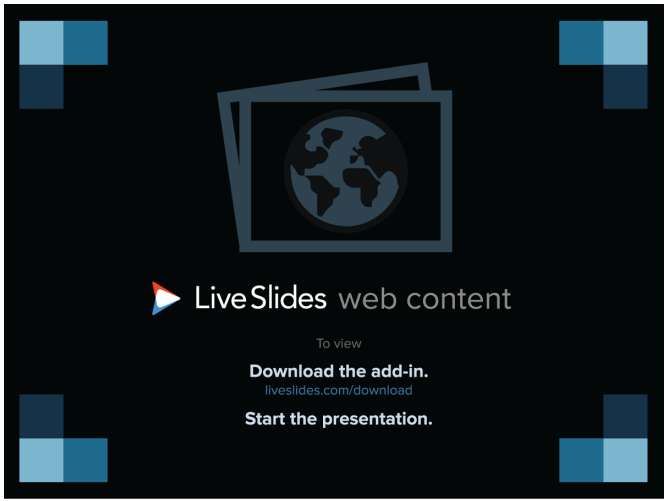
Cálculo del comando de giro

- Obtener los valles (zonas a 0)
- De todos ellos, seleccionar el valle más cercano a la dirección actual y que sea suficientemente ancho



VFH vs VFF

- El VFH soluciona razonablemente los problemas de **inestabilidad** del VFF en zonas estrechas
- Tampoco requiere de suavizado de la velocidad angular



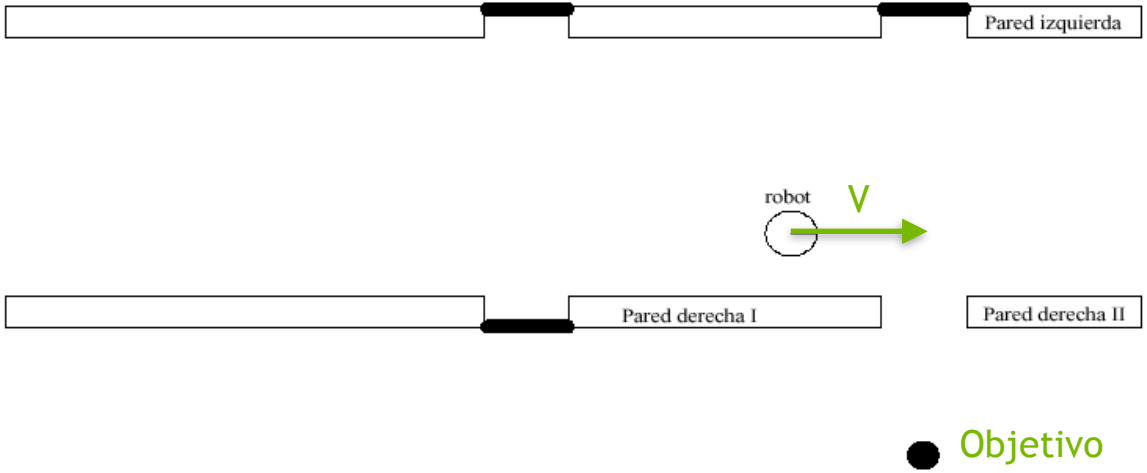
https://www.youtube.com/watch?v=ZhtLo08pM1s

Índice

Introducción a la evitación de obstáculos
Métodos de campo de fuerzas
Método de la ventana dinámica

Métodos de control en el espacio de velocidades

Los anteriores métodos no tenían en cuenta las restricciones dinámicas del robot: velocidad y aceleración. **¿Qué sucedería si la V del robot fuera alta e intentáramos un comando de giro hacia el objetivo?**



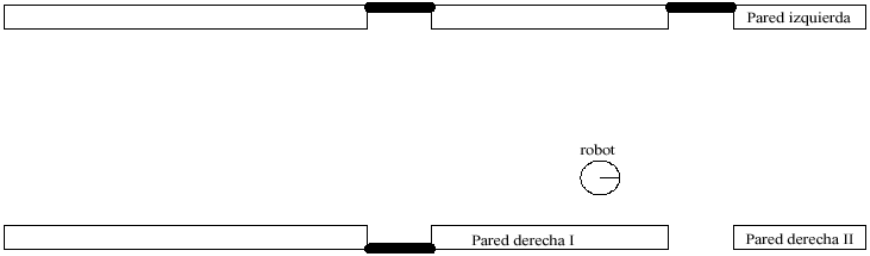
Problema: los métodos al estilo VFF/VFH **separan** el cálculo de la dirección óptima de la generación del comando de movimiento.

Solo podríamos asegurar seguir la dirección óptima con una aceleración infinita, lo cual no es el caso…

Método de la ventana dinámica

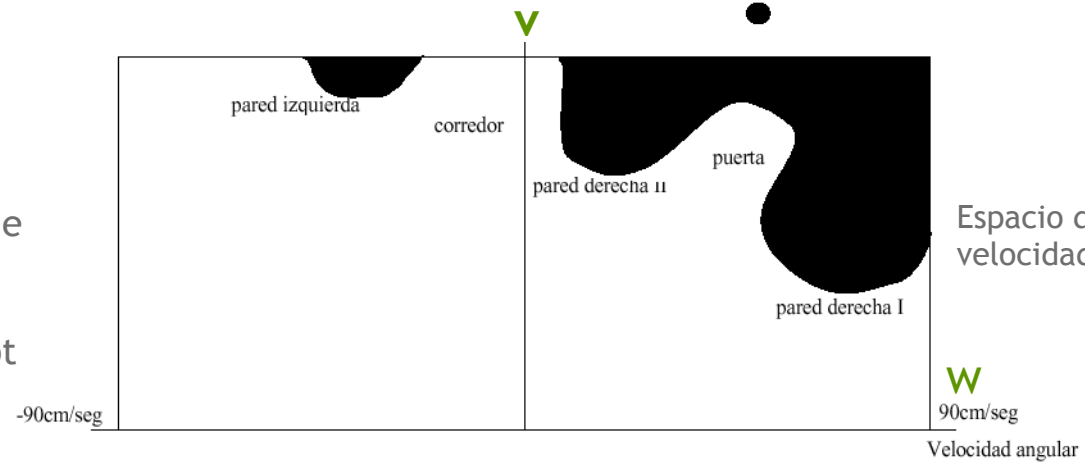
- Dynamic Window Approach (DWA) Fox, Burgard, Thrun, 1997
- Trabaja en el **espacio de velocidades**, no en el espacio de coordenadas
- Espacio de velocidades: conjunto de velocidades (lineales y rotacionales) que puede alcanzar el robot (v, w)
  - El rango de v y w viene impuesto por las especificaciones del robot

Espacio de velocidades



Espacio de coordenadas

El rango total para v y w viene dado por las características físicas del robot



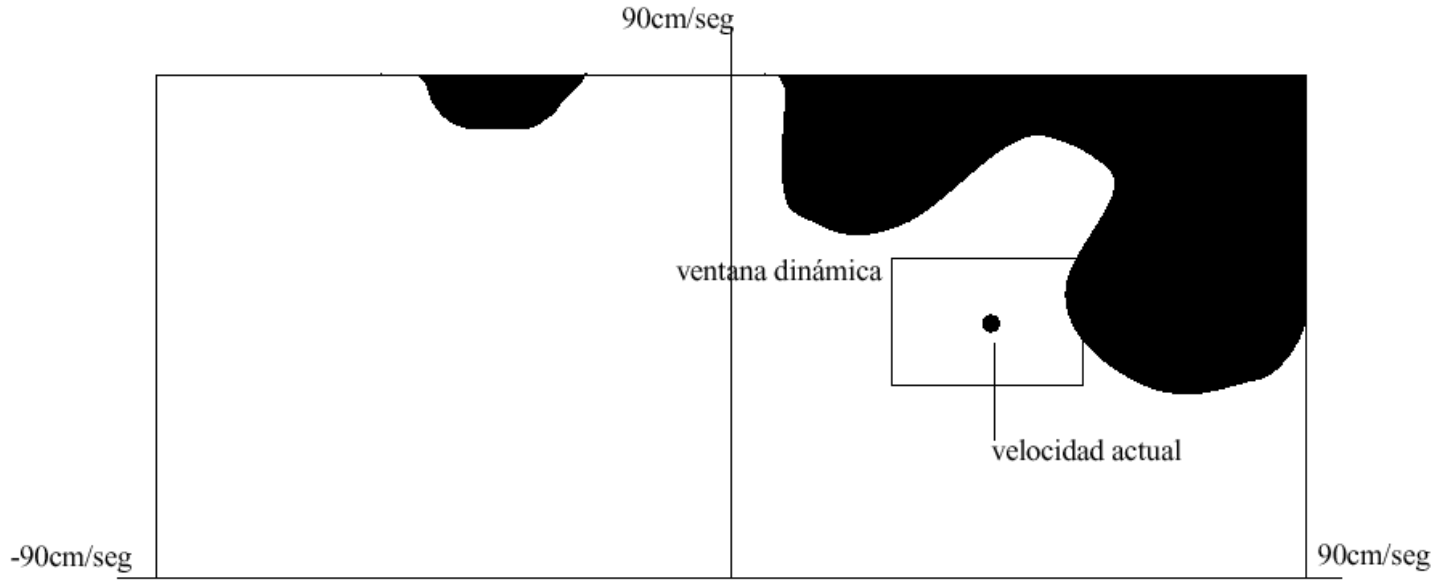
Espacio de velocidades

Búsqueda en el espacio de velocidades

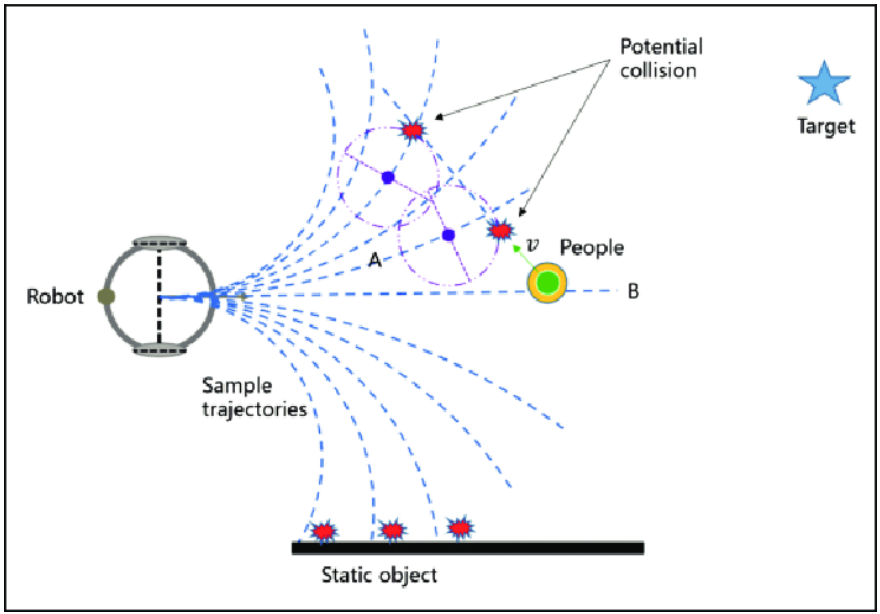
- Debemos encontrar **velocidades admisibles**: el robot consigue parar antes de chocar con un obstáculo (zonas negras)

- Debido a la **inercia** (aceleraciones posibles) sólo debemos buscar en un subespacio alrededor de la velocidad actual (**ventana dinámica**)

Ventana dinámica



- Un **punto** en el espacio (v,w) equivale a una **trayectoria semicircular** en el "mundo real" (si v, w se mantienen ctes. en un intervalo $\triangle t$)



Para hacer factible la búsqueda solo nos preocuparemos del **siguiente instante** de tiempo, asumiendo que a partir de ahí la velocidad es cte.

Buscar la velocidad óptima para cada instante por adelantado nos daría un espacio de búsqueda **exponencial**

Objetivos

Queremos que la (v,w) elegida para el siguiente $\triangle t$:

- Genere un movimiento **hacia el objetivo**
- Se acerque lo menos posible a los **obstáculos**
- La v sea la **mayor posible**

Como es lógico, estos objetivos a veces serán **contradictorios**

Formalización: función objetivo

- Queremos maximizar

$$G(v,w) = \alpha \cdot heading(v,w) + \beta \cdot dist(v,w) + \gamma \cdot vel(v,w)$$

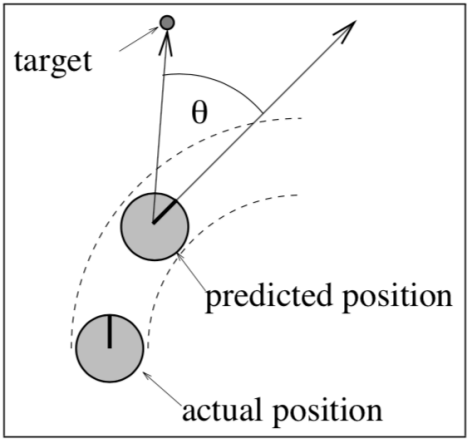
- donde:

  α, β, γ permiten dar más peso a una u otra función

Heading(v, w)

Mide la **alineación** de la dirección que lleva el robot con la que le llevaría al objetivo

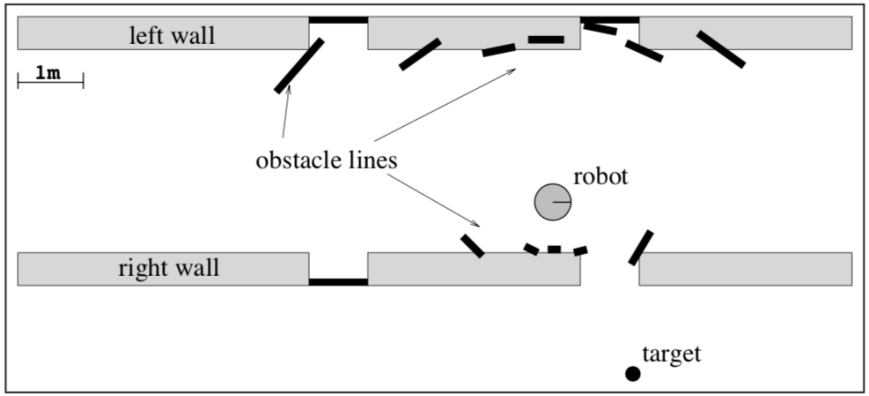
- Se calcula la posición y orientación que tendría el robot una vez aplicadas las velocidades v, w
- Se calcula el ángulo θ que habría que girar al robot para que vaya hacia el objetivo
- La función devuelve 180º-θ (así, valores altos representan θ bajos)

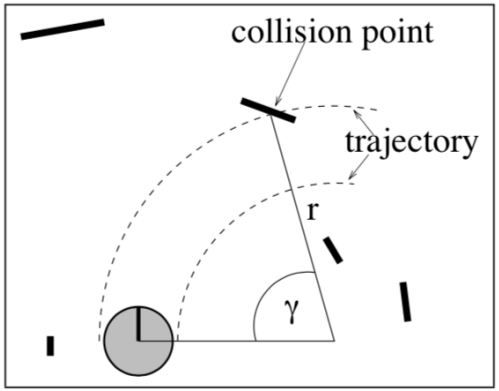


Dist (v, w)

Distancia al obstáculo más cercano que intersecta con esa curvatura

- Los obstáculos detectados por los sonares se modelan como segmentos perpendiculares al eje del sensor, y con tamaño igual al cono del sonar a esa distancia



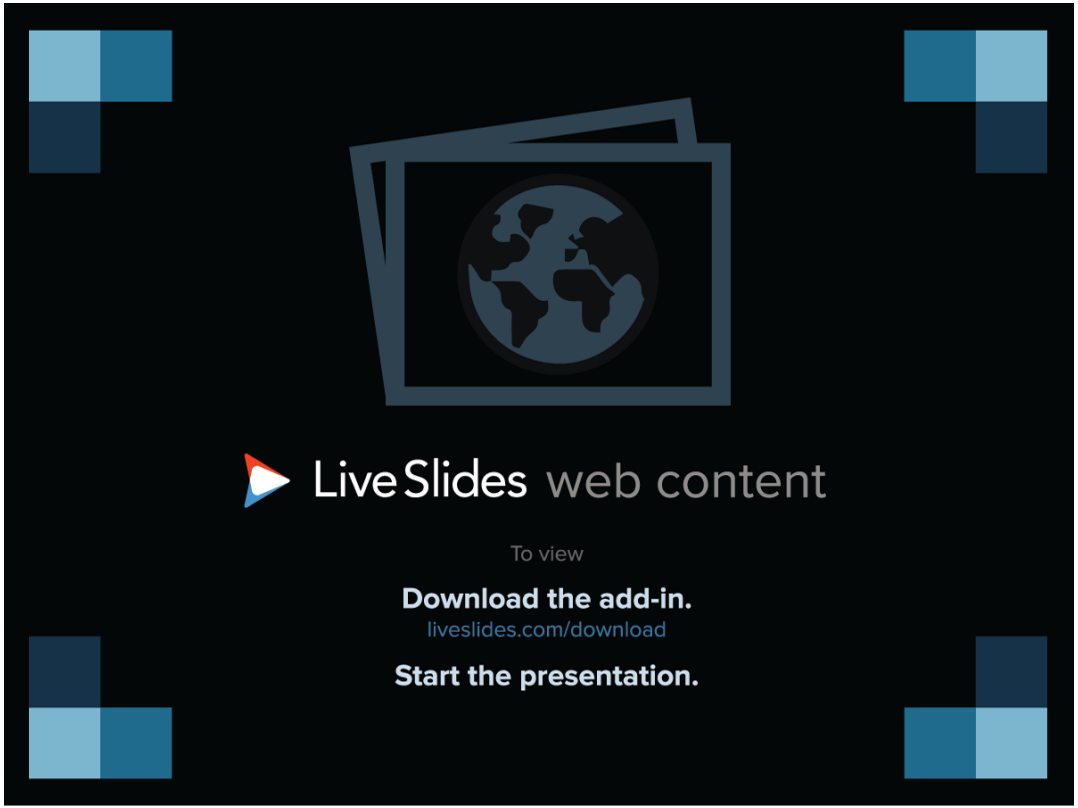


Vel(v, w)

Simplemente devuelve la **velocidad lineal** v

Como estamos maximizando, “premiaremos” velocidades altas

Aplicación: Rhino, el robot “guía de museo” (1997)

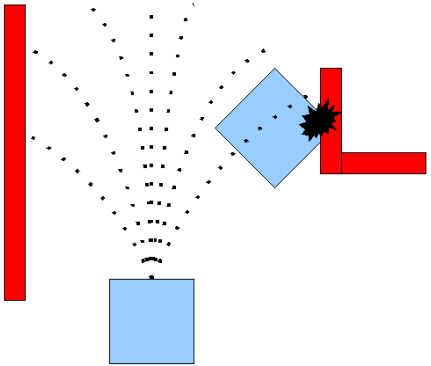


https://www.youtube.com/watch?v=lT-RF8TKRm0

ROS

DWA es el algoritmo **usado por defecto en ROS** para evitar obstáculos

Implementado en el paquete dwa_local_planner



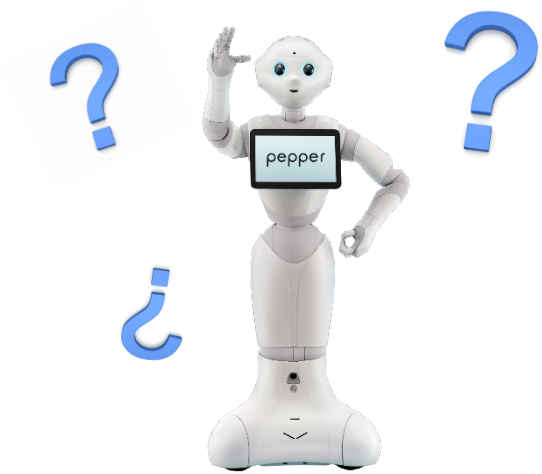
http://wiki.ros.org/dwa_local_planner

Problemas de la ventana dinámica

El robot no frena a tiempo para entrar por la puerta y tiene que retroceder. Para frenar a tiempo **habría tenido que planificar en un intervalo mayor** que solo el siguiente instante

Robots Móviles

Grado en Ingeniería Robótica



otto @ dccia.ua.es