

PROGRAMACIÓN DE ROBOTS

**Robots móviles:
navegación**

Robots móviles: navegación

- 1.- Navegación de robots móviles
- 2.- Navegación deliberativa
- 3.- Navegación reactiva

Robots móviles: navegación

- 1.- Navegación de robots móviles
- 2.- Navegación deliberativa
- 3.- Navegación reactiva

1.-Navegación de robots móviles

Bibliografía

Braitenberg V. (1994), *Vehicles: experiments in synthetic psychology*, The MIT Press.

Corke P. (2011), *Robotics, vision and control: fundamentals algorithms in MATLAB*, Springer.

Hansen J. (2007), *Lego NXT power programming : robotics in C*, Variant Press.

Jones J.L. (2004), *Robot Programming. A Practical Guide to Behaviour-Based Robotics*, McGraw-Hill.

Mataric M. J. (2007), *The Robotics Primer*, The MIT Press.

Siciliano B y Khatib O. (2008), *Handbook of Robotics*, Springer.

1.-Navegación de robots móviles

Bibliografía

Inteligencia Artificial: Russell S., Norvig P. (2009), *Artificial Intelligence, A Modern Approach*, Prentice Hall.

Cinemática y localización: Siegwart R, Nourbakhsh I, Scaramuzza D. (2011), *Introduction to Autonomous Mobile Robots*, The MIT Press.

Navegación: Choset H., Lynch K. M., Hutchinson S., Kantor G., Burgard W., Kavraki L. E., Thrun S. (2005), *Principles of Robot Motion: Theory, Algorithms, and Implementations*, A Bradford Book.

Navegación: LaValle S. M. (2006), *Planning Algorithms*, Cambridge University Press.

1.-Navegación de robots móviles

Bibliografía

Martínez Rodríguez J. L. (1995), *Seguimiento Automático de Caminos en Robots Móviles*, Tesis Doctoral, Universidad de Málaga.

Muñoz Martínez, V. F. (1997), *Planificación de Trayectorias para Robots Móviles*, Tesis Doctoral, Universidad de Málaga.

Mandow Andaluz A. (1999), *Navegación de Robots Móviles Basada en Comportamientos*, Tesis Doctoral, Universidad de Málaga.

Cruz Martín A. (2004), *Planificación de Trayectorias en Sistemas Multirrobot*, Tesis Doctoral, Universidad de Málaga.

1.-Navegación de robots móviles

Definición de navegación

- Corke: “Robot navigation is the problem of guiding a robot to its goal”
- Mataric: “Navigation refers to the way a robot finds its way in the environment”
- Cruz-Martín: “Navegar consiste básicamente en moverse a través de un entorno evitando posibles colisiones con elementos del mismo, tanto estáticos como dinámicos”

1.-Navegación de robots móviles

Definición de navegación

- Cuestiones básicas a partir de las definiciones anteriores:
 - Movimiento
 - Objetivo
 - Seguridad
 - La navegación es una tarea fundamental en cualquier robot móvil (terrestre, aéreo o subacuático), independientemente de su estructura física y su sistema de locomoción.
 - Quizás es el problema más antiguo y más estudiado en Robótica, incluyendo sus subproblemas (localización, SLAM, coverage...)
-

1.-Navegación de robots móviles

Definición de navegación

- ¿Cómo navegamos los humanos?
 - Nos basamos en mapas y señales
 - No todos tenemos la misma capacidad de navegación
- Navegación con mapas:
 - Hay una representación interna
 - Hay un plan
 - No necesito percibir el objetivo
 - Necesito saber dónde estoy
- ¿Se puede navegar sin mapa, es decir, sin representación interna y sin plan?
 - Sí: hormigas, abejas... No tienen suficientes neuronas para ello.
 - En Robótica: navegación reactiva
 - Dirigirse a una fuente de luz
 - Coverage sin mapa: movimiento aleatorio

Como la Zoomba que teníamos
que daba vueltas de forma errática
durante 1 hora o así

Robots móviles: navegación

- 1.- Navegación de robots móviles
- 2.- Navegación deliberativa
- 3.- Navegación reactiva

2.-Navegación deliberativa

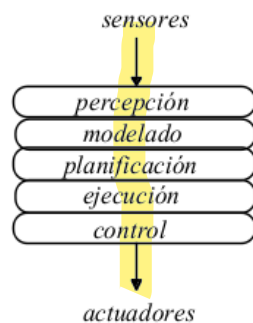
Conceptos

- Definición
- Ventajas y desventajas
- Jerarquía de movimientos
- Generación de caminos
- Generación de trayectorias
- Generación de rutas
- Técnicas de seguimiento
- Navegación multirrobot

2.-Navegación deliberativa

Definición

- En la navegación deliberativa (planificada o estratégica) existe un plan previo al movimiento.
- Ese plan se basa en los datos detectados por los sensores.
- El esquema tradicionalmente seguido es el siguiente:



- Justificación: primera aproximación histórica al problema de la navegación, muy influenciado por la inteligencia artificial de la época (60-70's), basada en la planificación de tareas. Ejemplo: NavLab

2.-Navegación deliberativa

Ventajas y desventajas

Ventajas

- Tiene memoria y representación interna, por lo que el objetivo final no se pierde.
- Puede ofrecer la mejor solución.

Desventajas

- O el entorno es completamente estático, o no es capaz de adaptarse a los cambios.
- Elevado coste computacional (tiempo, memoria), a pesar de los avances en hardware.

Ésta mierda es lentísima

2.-Navegación deliberativa

Jerarquía de movimientos

- Definiciones:

- Camino: conjunto de posiciones en el espacio que debe seguir un robot. Debe ser:
 - Seguro (libre de obstáculos)
 - Admisible para la cinemática y dinámica del vehículo.
- Trayectoria: camino unido a un perfil de velocidad.
- Ruta: conjunto de puntos de referencia que debe alcanzar un robot para cumplir una misión. Un camino se define entre dos puntos de una ruta.

- Esta es mi filosofía, pero podéis encontrar otras, no es un tema cerrado.

- Todo el planteamiento del tema está pensado para 2D.

2.-Navegación deliberativa

Jerarquía de movimientos

Por tanto, hay dos etapas secuenciales:

- **Generación** (caminos, trayectorias, rutas): establecer los puntos que hay que visitar (y sus velocidades, en el caso de las trayectorias).
- **Seguimiento** (*path tracking*): lograr que el robot físicamente se desplace a esos puntos previamente calculados.

2.-Navegación deliberativa

Generación de caminos

- El generador de caminos nos proporcionará un conjunto de posturas o poses.
- Cada postura es una tupla (x,y,θ) a la que se le puede añadir la curvatura K , dependiendo del espacio de configuraciones usado.
- El camino generado debe presentar las siguientes características:
 - Eliminar omnidireccionalidad.
 - Considerar las características cinemáticas y dinámicas del vehículo, debido a la enorme influencia que tienen sobre la capacidad del robot a la hora de seguir el camino.
 - Lograr el menor error posible cuando se siga el camino.
 - No atravesar obstáculos.

2.-Navegación deliberativa

Generación de caminos

- Existen muchos métodos de generación de caminos:
 - **Concatenación de segmentos:** es el método más simple, pero únicamente asegura la continuidad en posición.
 - **Concatenación de rectas y arcos circulares;** se garantiza la continuidad en posición y orientación, pero no en curvatura.
 - **El camino producido contiene curvas clotoïdales o espirales cúbicas.** Éstas sí aseguran la continuidad en curvatura, pero no tienen una forma cerrada, lo que hace su cálculo muy costoso computacionalmente, y por tanto son poco aplicables en tiempo real
 - **El camino se genera usando polinomios de 5º grado, polinomios de Bezier o curvas polinómicas polares.** Son caminos constantes en curvatura, pero debido a la unión de segmentos polinómicos, la transición de curvatura no es siempre suave.
 - **El camino se genera usando curvas paramétricas del tipo B-spline o β -spline.** Además de asegurar curvatura constante, ambas tienen una forma cerrada y pueden generarse fácilmente usando algoritmos eficientes.

2.-Navegación deliberativa

Esta distinción es importante:
Un camino NO tiene componente
velocidad,
una trayectoria SÍ.

Generación de trayectorias

En el god forsaken idioma que es el
inglés se usan indistintamente path y
trajectory, ojo con eso

- Se denomina trayectoria a la unión de un camino y un perfil de velocidad.
- Por tanto, combinamos una planificación espacial (camino) con una planificación temporal (velocidad)
- Ventaja: sabemos a qué velocidad hay que recorrer cada punto del camino, en lugar de suponer que se va a recorrer a una velocidad baja. Puede ser importante si:
 - El camino tiene tramos complejos (vuelco en curvas cerradas)
 - Hay obstáculos móviles (una trayectoria es segura o no dependiendo de la velocidad)
- La trayectoria está compuesta de posturas temporales, resultado de unir una componente de velocidad v_i y aceleración a_i a cada postura del camino.

2.-Navegación deliberativa

Generación de trayectorias

- La planificación temporal que hagamos será el resultado de combinar una serie de restricciones de velocidad (y aceleración)
- Las limitaciones de velocidad y aceleración se obtienen a partir de los modelos cinemáticos y dinámicos del vehículo.
- Los valores v_i y a_i de cada tupla se calculan como el valor mínimo de todas las restricciones de velocidad y aceleración que afectan a la postura i-ésima asociada:

$$v_i = \text{MIN}(vr_{i0}, vr_{i1}, \dots, vr_{ip}) \quad a_i = \text{MIN}(ar_{i0}, ar_{i1}, \dots, ar_{iq})$$

donde MIN se considera la función encargada de seleccionar el mínimo de los valores vr_{ij} y ar_{ij} que representan las limitaciones de la postura i-ésima.

2.-Navegación deliberativa

Generación de trayectorias

- Las restricciones de velocidad y aceleración se clasifican en:
 - Físicas
 - De construcción
 - Cinemáticas
 - Dinámicas
 - Operacionales

2.-Navegación deliberativa

Generación de trayectorias

- Restricciones físicas:

- De construcción: establecidas por el sistema de locomoción del propio vehículo, que fijan una velocidad y aceleración máximas.
- Cinemáticas: dependientes del modelo cinemático utilizado para representar el robot. Fijan la velocidad y aceleración máxima de algunos puntos destacados del modelo respecto a la velocidad de cada una de las ruedas.
- Dinámicas: motivadas por el comportamiento dinámico del vehículo y por las fuerzas que actúan sobre el mismo desviándolo de su camino:
 - Máximo rango de variación de la curvatura
 - Interacción de las ruedas del vehículo con el suelo (deslizamientos). ¡Pueden depender incluso del tipo de suelo :S!

2.-Navegación deliberativa

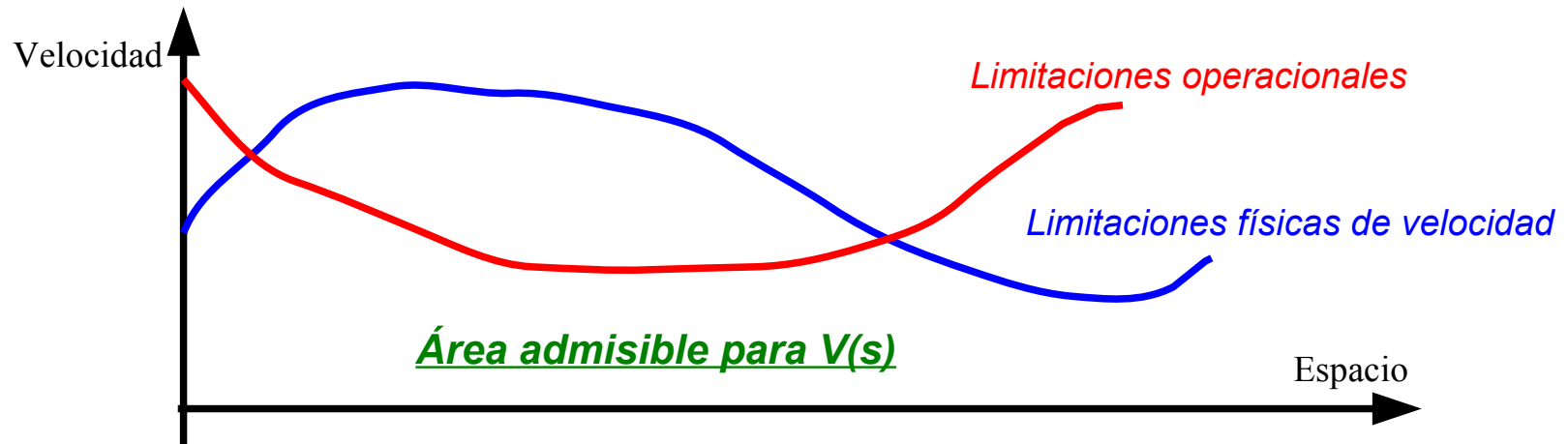
Generación de trayectorias

- Restricciones operacionales:
 - Debidas a la interacción del vehículo con su entorno:
 - Distancia al punto de parada (objetivo u obstáculo ineludible)
 - Restricciones de navegación

2.-Navegación deliberativa

Generación de trayectorias

- Plano de fase espacio-velocidad: permite representar el perfil de velocidad deseado junto con las restricciones de velocidad que hay que cumplir



- Metodologías de planificación de velocidades:
 - Simultáneamente con el camino
 - Primero el camino, luego el perfil temporal

2.-Navegación deliberativa

Generación de rutas

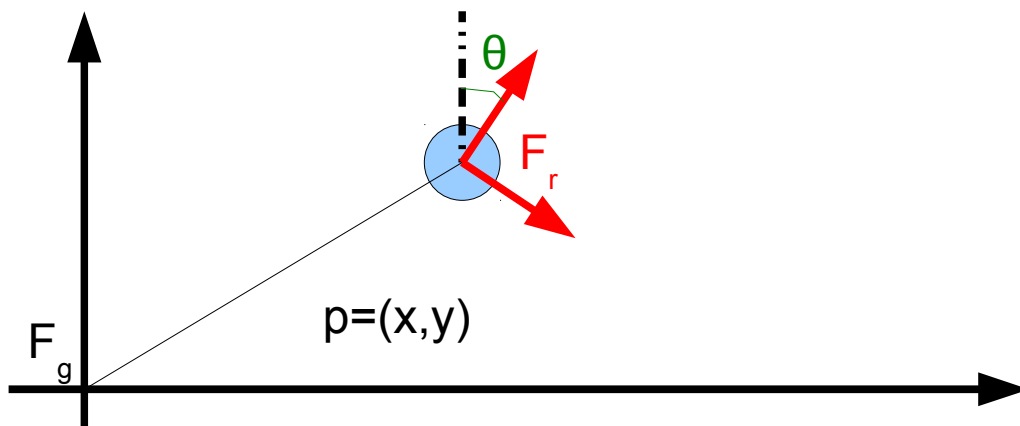
- A la hora de generar una ruta, ésta debe ser segura:
 - Continua al menos en posición (supone por tanto que el vehículo es omnidireccional)
 - Libre de obstáculos
 - No se preocupa de características cinemáticas y dinámicas (tarea del generador de caminos)

2.-Navegación deliberativa

Generación de rutas

- Formalización del problema

- Configuración q de un robot R : vector cuyas componentes proporcionan información completa sobre el estado actual del mismo.



- $q=(p,\theta) = (x,y,\theta)$
- Espacio de configuraciones C del robot R : todas las configuraciones que puede tomar el robot en su entorno.

2.-Navegación deliberativa

Generación de rutas

- Formalización del problema

- C espacio de configuraciones q
- $R(q)$ es el subconjunto de C que ocupa el robot R cuando se encuentra en q : $R(q)=\{q_i \in C / ||q, q_i|| \leq \rho\}$
- Los obstáculos que puedan aparecer en el espacio de configuraciones se definen como $B=\{b_1, b_2, \dots, b_q\}$

2.-Navegación deliberativa

Generación de rutas

- Formalización del problema

- El conjunto de configuraciones de C ocupadas por un obstáculo se define por $b_i(q)$, y por tanto, el espacio libre de obstáculos se establece como

$$C_l = \{q \in C / R(q) \cap \left(\bigcup_{i=1}^q b_i(q) \right) = \emptyset\}$$

- La planificación de una ruta se traduce entonces en la búsqueda de una sucesión de posturas tal que la primera sea la postura actual del robot q_a y la última sea la postura objetivo q_f : $Q_r = \{q_a, \dots, q_f / q_i \in C_l\}$

2.-Navegación deliberativa

Generación de rutas

- Métodos de generación de rutas:

- Grafos de visibilidad
- Descomposición en celdas
- Distancia transformada
- D^*
- Diagramas de Voronoi
- Mapa probabilístico
- RRT

2.-Navegación deliberativa

Generación de rutas

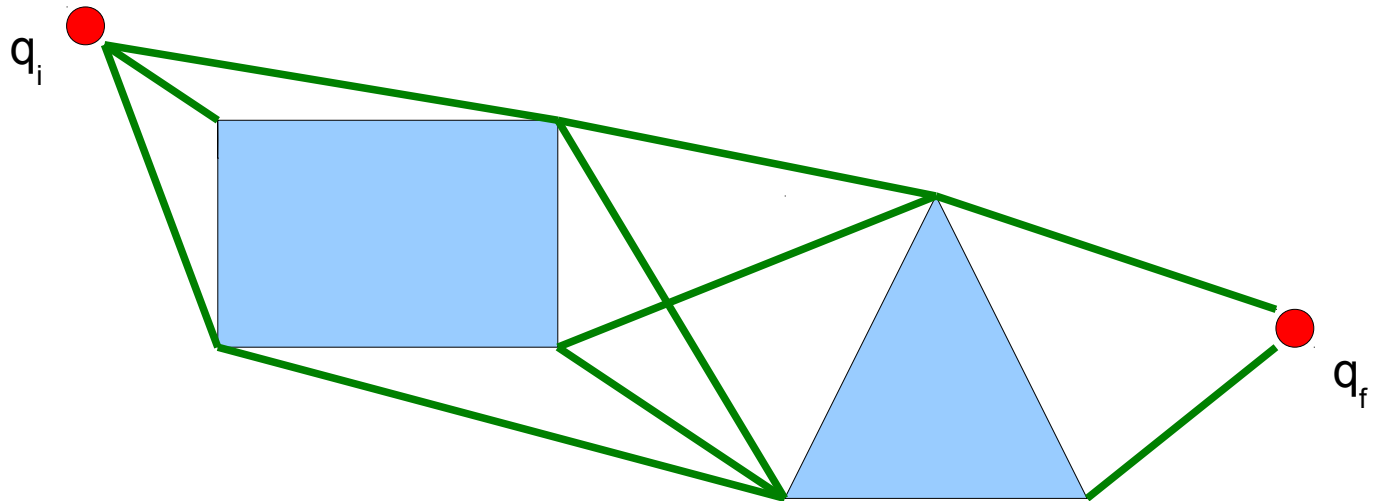
- Grafos de visibilidad

- Los obstáculos se modelan como polígonos.
- Se construye un grafo
 - Nodos: vértices de los obstáculos+postura de inicio+postura de fin.
 - Aristas: unen nodos *visibles*, es decir, que pueden unirse por un segmento rectilíneo que no intersecta con ningún obstáculo.
- Se aplica un algoritmo de búsqueda en grafos que proporcione una ruta de coste mínimo según algún criterio.
- Bajo coste computacional
- Supone que el robot es un punto: hay que “inflar” los obstáculos

2.-Navegación deliberativa

Generación de rutas

- Grafos de visibilidad



2.-Navegación deliberativa

Generación de rutas

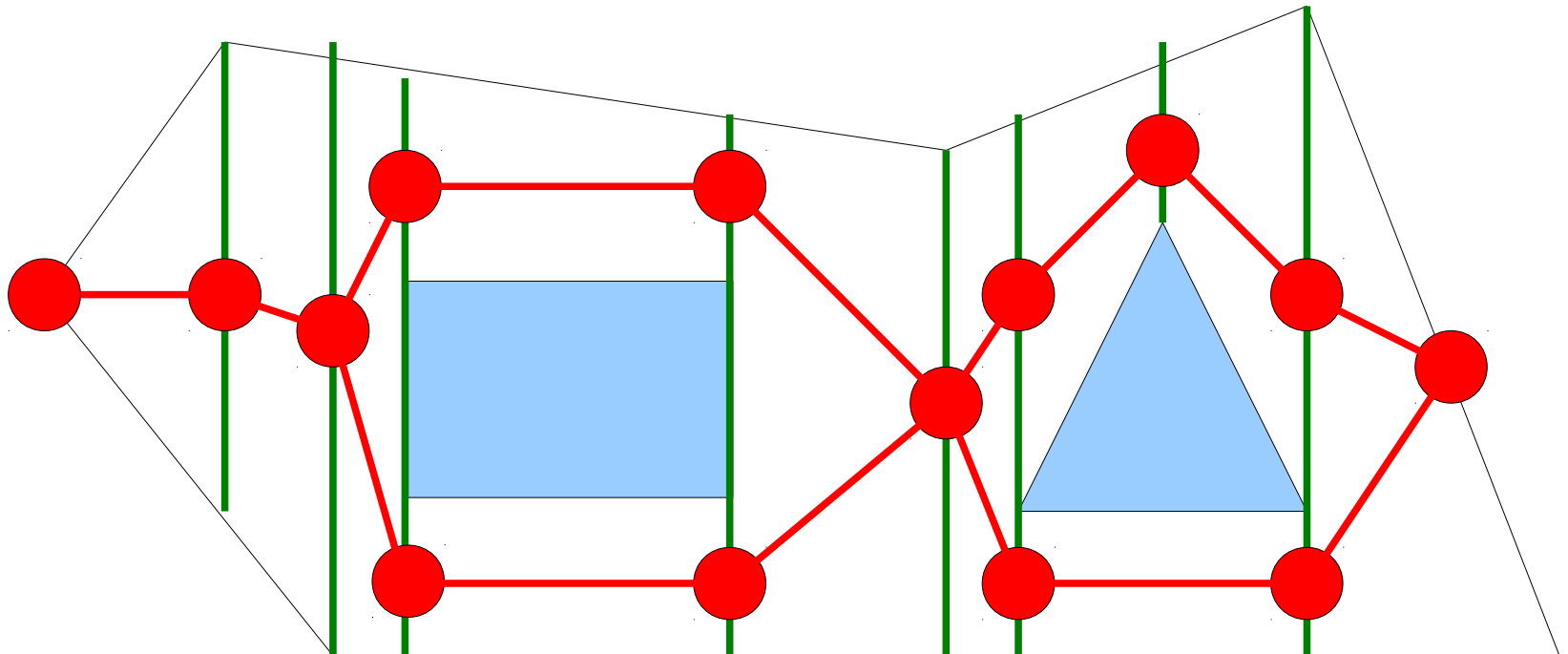
- Descomposición en celdas

- La idea es descomponer el espacio libre en celdas. La ruta construida es una sucesión continua de celdas, de forma que la postura inicial esté en la primera celda, y la postura final en la última.
- Son necesarios tres pasos:
 - Descomposición del espacio libre en celdas. Hay muchas posibilidades, pero debe ser un método sencillo y que cubra todo el espacio.
 - Generar un grafo de conectividad, cuyos nodos representen a las celdas.
 - Aplicar un algoritmo de búsqueda en el grafo.

2.-Navegación deliberativa

Generación de rutas

- Descomposición en celdas: descomposición trapezoidal



2.-Navegación deliberativa

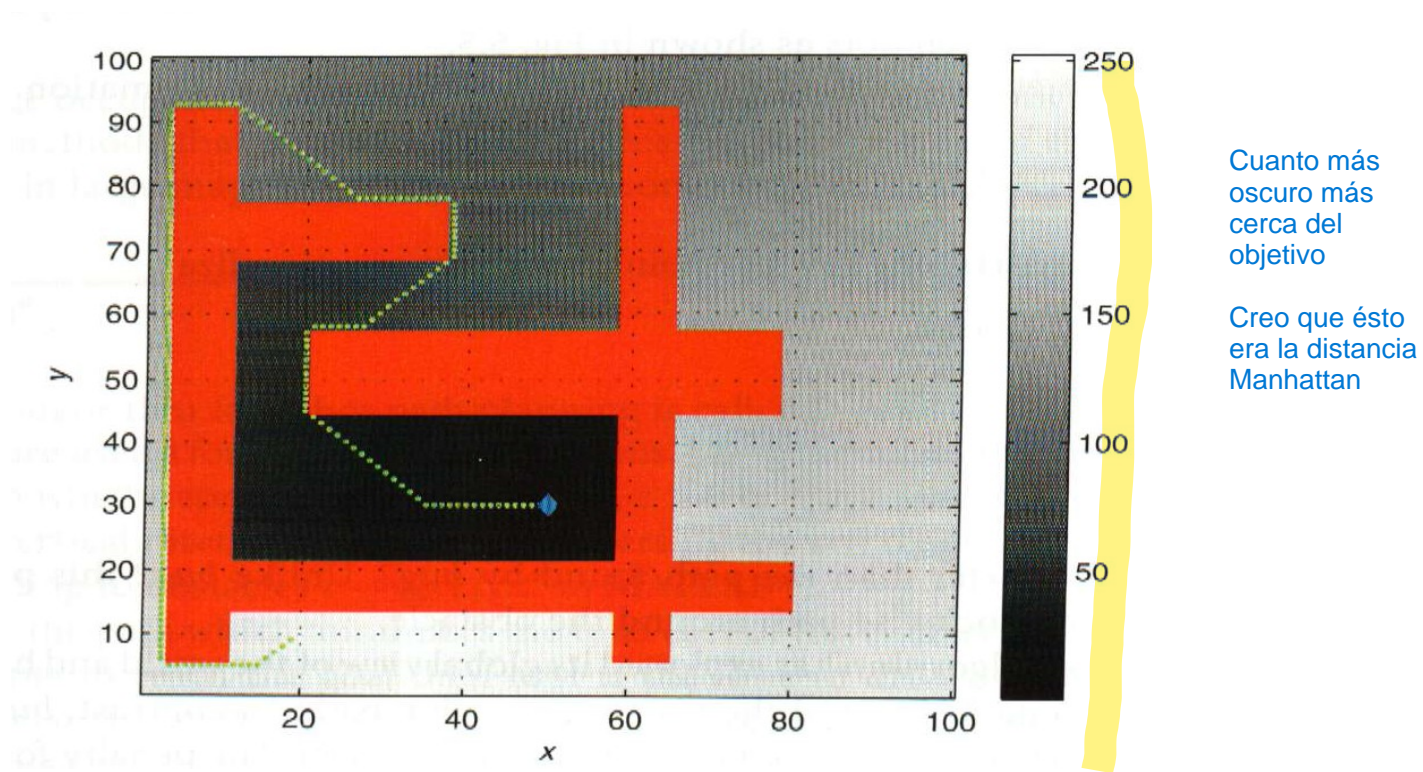
Generación de rutas

- Distancia transformada (1986; aplicación robótica, 1988)
 - Usamos una rejilla de ocupación (celdas ocupadas o libres), no hay restricciones holonómicas, y el robot conoce su posición.
 - Se construye una matriz de distancias transformadas del mismo tamaño que el mapa original, pero cuyas celdas contienen los valores de distancia al objetivo (sorteando obstáculos si es necesario):
 - Distancia euclídea, u otras (Manhattan: $|\Delta x| + |\Delta y|$)
 - Algoritmo: moverse sobre esa matriz, buscando siempre la celda con la distancia más corta. Por tanto, tiene una visión global de todo el mapa.
 - Hay que crear esa matriz, y eso es costoso computacionalmente: $O(N^2)$

2.-Navegación deliberativa

Generación de rutas

- Distancia transformada (1986; aplicación robótica, 1988)



Corke, *Robotics, Vision and Control. Fundamental Algorithms in Matlab*

2.-Navegación deliberativa

Generación de rutas

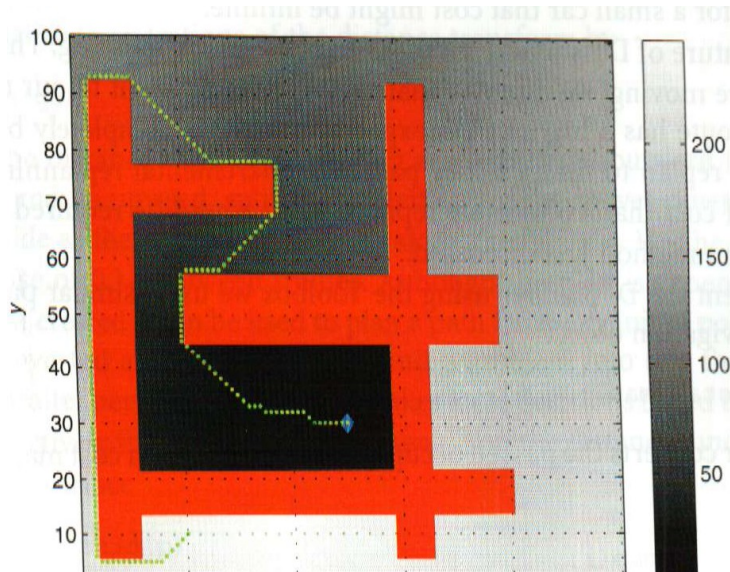
- D* (1994)

- D* = Dynamic A*, es un A* en el que los costes de los arcos pueden variar cuando se ejecuta el algoritmo; varía en que se mueve hacia atrás.
- Extensión de A* muy útil para aplicaciones reales; muy usado en robótica móvil y navegación.
- Hay diferentes variantes.
- Permite definir nuestra propia función de coste (normalmente distancia, pero no tiene por qué ser así siempre: evitar terrenos complicados, giros difíciles...)
- Permite replanificaciones: el camino puede recalcularse aunque se modifique el mapa de coste.
- Sigue siendo computacionalmente costoso.

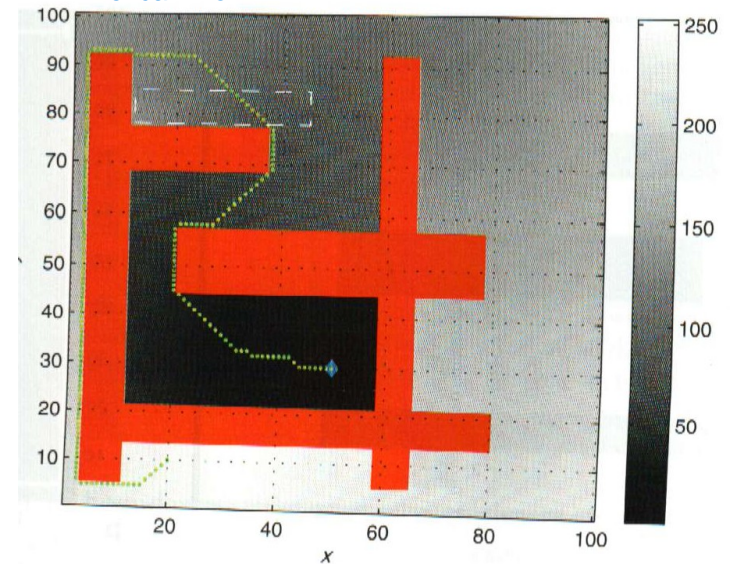
2.-Navegación deliberativa

Generación de rutas

- D* (1994)



La franja de gris del rectángulo está ligeramente más claro, es decir, ha cambiado el escenario y se recalcula el camino



Corke, *Robotics, Vision and Control. Fundamental Algorithms in Matlab*

2.-Navegación deliberativa

Generación de rutas

- Diagramas de Voronoi
 - Los métodos anteriores tienen una fase de preparación costosa, y una fase de consulta simple.
 - La fase de preparación depende del objetivo: si el objetivo varía, hay que repetirla
 - En este caso planteamos un nuevo enfoque: mapa de carreteras o roadmap (ejemplo, red ferroviaria).
 - En el caso de los diagramas de Voronoi, se construye el roadmap manteniendo siempre la equidistancia a los obstáculos.

2.-Navegación deliberativa

Generación de rutas

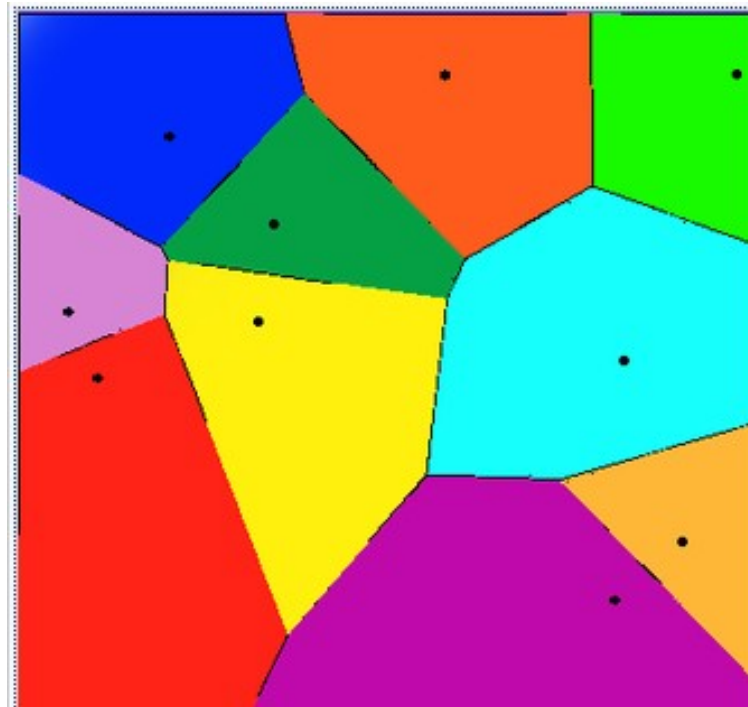
- Diagramas de Voronoi

- Se parte de un conjunto de puntos $\{p_1, \dots, p_n\}$ en el plano Euclídeo, llamados sites o seeds. Dado un site p_k , cada celda o región de Voronoi consiste en todos los puntos cuya distancia a p_k no es mayor que su distancia a cualquier otro site. Los segmentos del diagrama de Voronoi son todos los puntos del plano equidistantes a los dos sites más cercanos. Los nodos o vértices Voronoi son los puntos equidistantes a tres o más sites.

2.-Navegación deliberativa

Generación de rutas

- Diagramas de Voronoi



Los puntos negros son los sites o seeds.
Todo los puntos de la zona roja estarán más cerca del seed rojo que de otro, y así con todos los colores.
Los segmentos son areas donde hay sites a la misma distancia.

La gracia de usar eso es que cada site podrá ser un obstáculo y usar siempre los segmentos entre las distintas áreas, evitando siempre los obstáculos

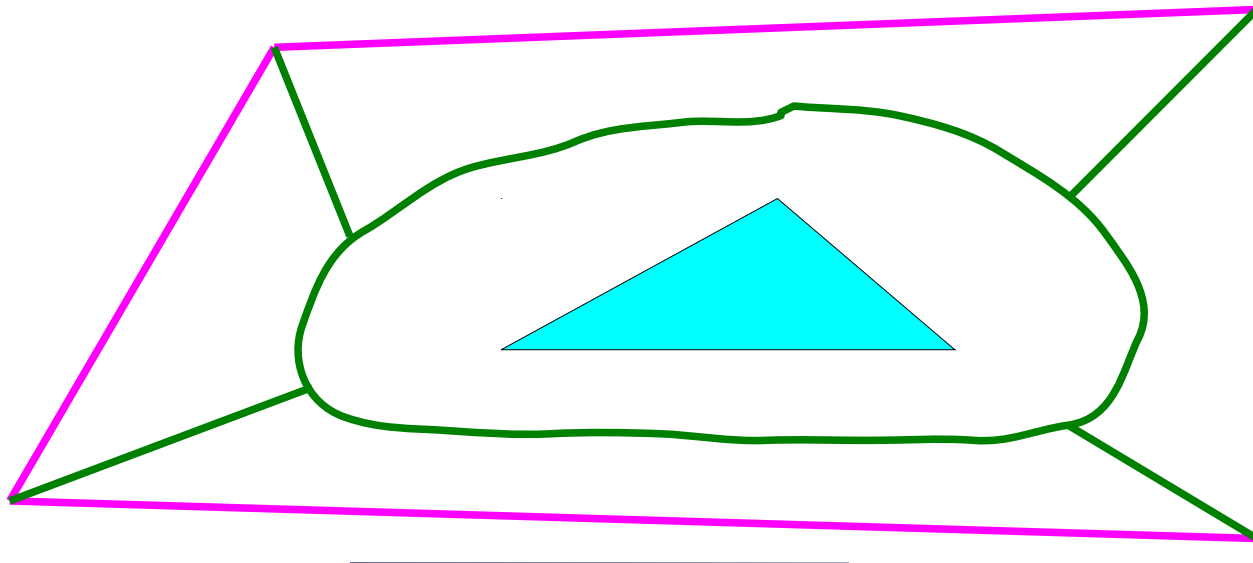
Wikipedia

2.-Navegación deliberativa

Generación de rutas

Diagramas de Voronoi

- ¿Cómo usarlo para generar rutas?
- Se busca la equidistancia entre obstáculos y límites del entorno.
- El roadmap surgido es una concatenación de segmentos rectilíneos y parabólicos.

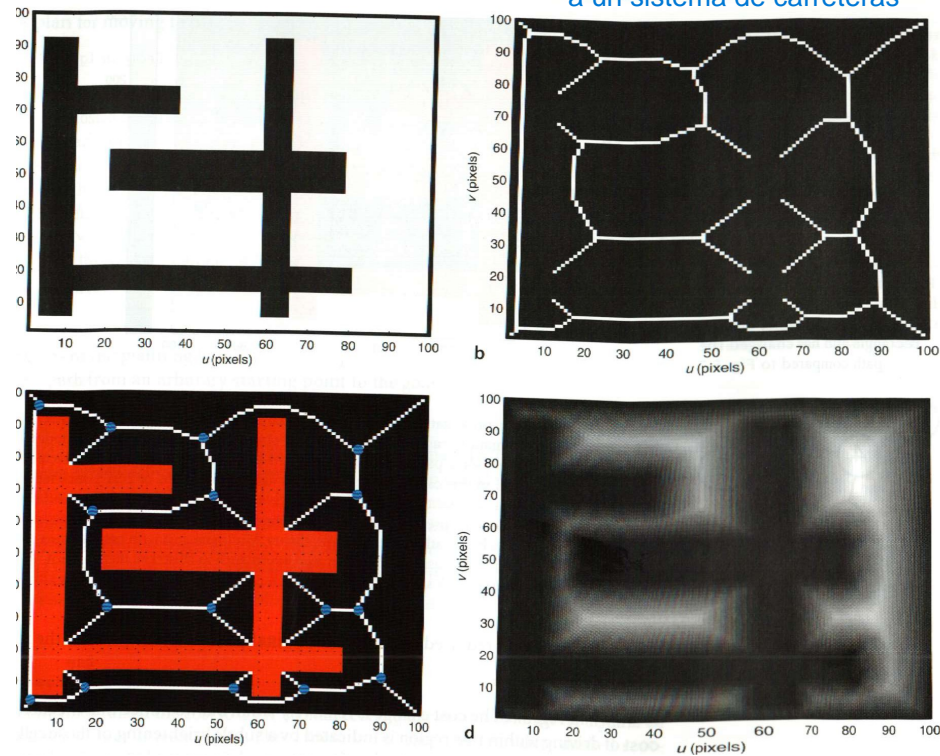


2.-Navegación deliberativa

Generación de rutas

Diagramas de Voronoi

Fíjate que los segmentos del diagrama de Voronoi es parecido a un sistema de carreteras



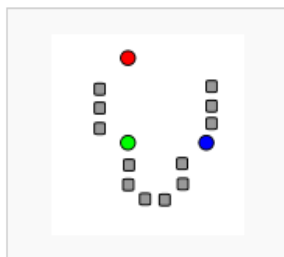
Corke, *Robotics, Vision and Control. Fundamental Algorithms in Matlab*

2.-Navegación deliberativa

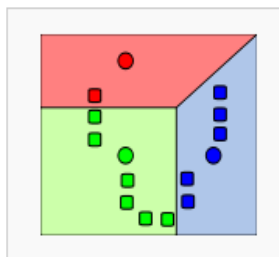
Generación de rutas

Diagramas de Voronoi

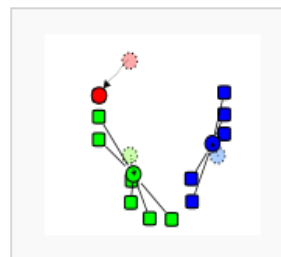
- Método muy utilizado para estudiar relaciones espaciales: geografía, geología, agricultura, GIS...
- VoroWiki: www.voronoi.com
- Relación algoritmo de clustering k-means/Voronoi: el algoritmo k-means genera una distribución de celdas Voronoi.



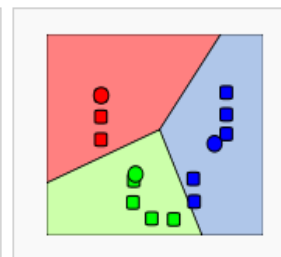
1) k initial "means" (in this case $k=3$) are randomly selected from the data set (shown in color).



2) k clusters are created by associating every observation with the nearest mean. The partitions here represent the Voronoi diagram generated by the means.



3) The centroid of each of the k clusters becomes the new mean.



4) Steps 2 and 3 are repeated until convergence has been reached.

2.-Navegación deliberativa

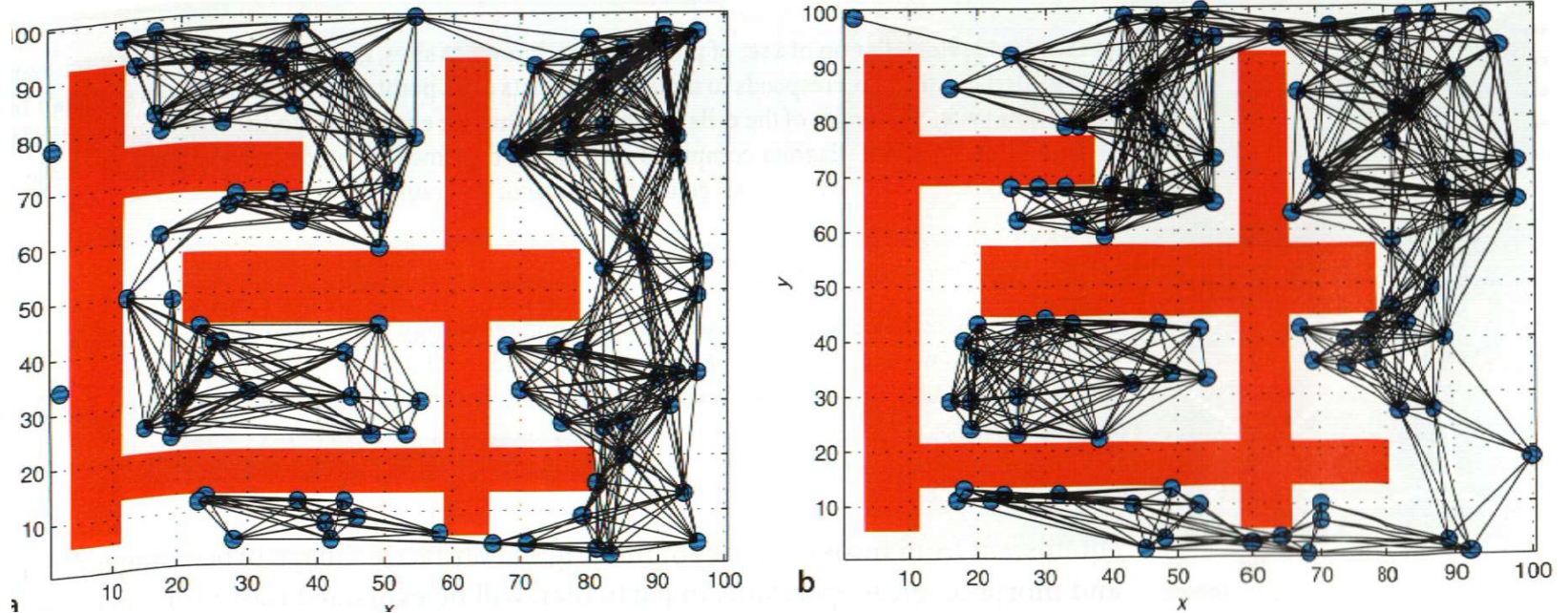
Generación de rutas

- Roadmap probabilístico (1996, muchas variaciones)
 - Los diagramas de Voronoi evitan el problema de modificar el objetivo, pero también son costosos.
 - En los roadmaps probabilísticos se muestrea aleatoriamente tomando N puntos del espacio libre:
 - Se forma un grafo con esos puntos, que no cruce obstáculos.
 - Coste de la arista: distancia entre nodos.
 - Ruta: recorro el grafo buscando siempre la arista menos costosa.
 - Problemas:
 - » Camino no óptimo.
 - » Pueden quedar nodos aislados.
 - » Pueden quedar zonas aisladas (y eso puede hacer que inicio y final no estén conectados)
 - » Pasillos estrechos poco explotados
 - » No hay una única solución

2.-Navegación deliberativa

Generación de rutas

- Roadmap probabilístico (1996, muchas variaciones)

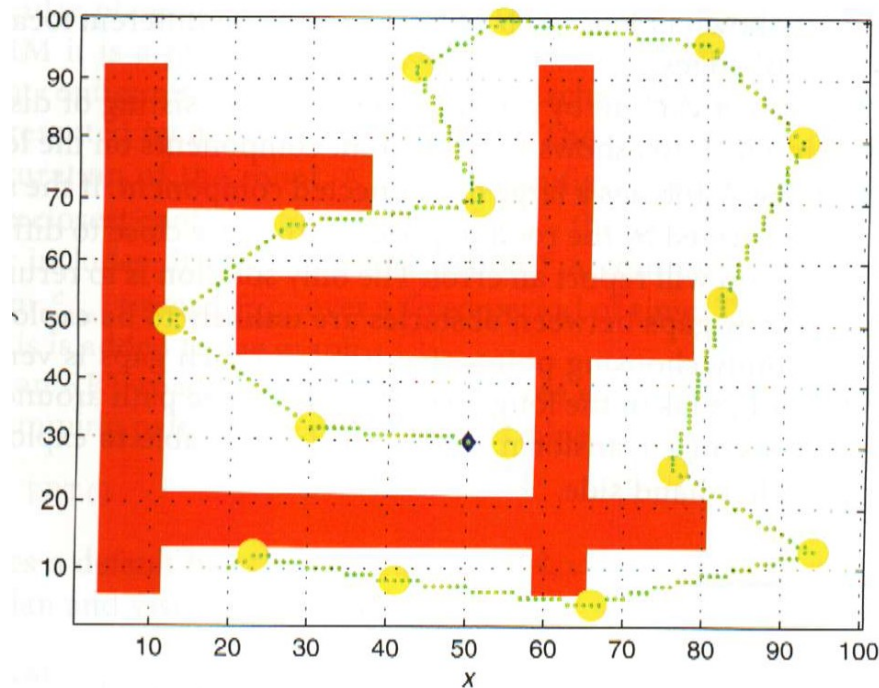


Corke, *Robotics, Vision and Control. Fundamental Algorithms in Matlab*

2.-Navegación deliberativa

Generación de rutas

- Roadmap probabilístico (1996, muchas variaciones)



Corke, *Robotics, Vision and Control. Fundamental Algorithms in Matlab*

2.-Navegación deliberativa

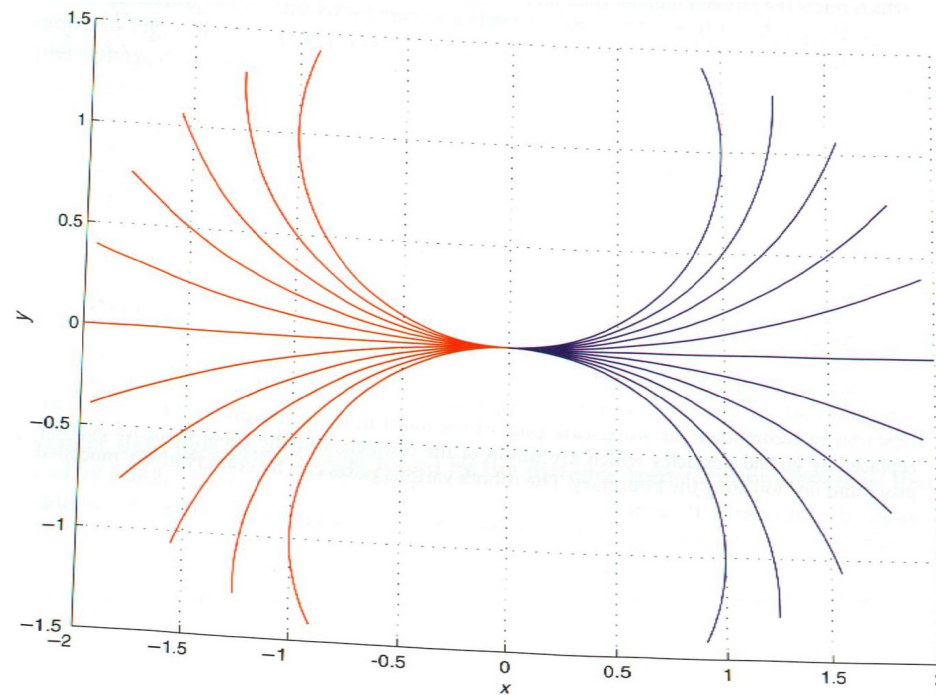
Generación de rutas

- RRT, Rapidly-exploring Random Tree
 - Sí tiene en cuenta el modelo de movimiento del vehículo, eliminando la omnidireccionalidad.
 - La idea es seleccionar un punto inicial, y obtener un conjunto de posturas alcanzables desde él. Si se repite el proceso, obtendríamos un gran conjunto de posturas posibles.

2.-Navegación deliberativa

Generación de rutas

- RRT, Rapidly-exploring Random Tree

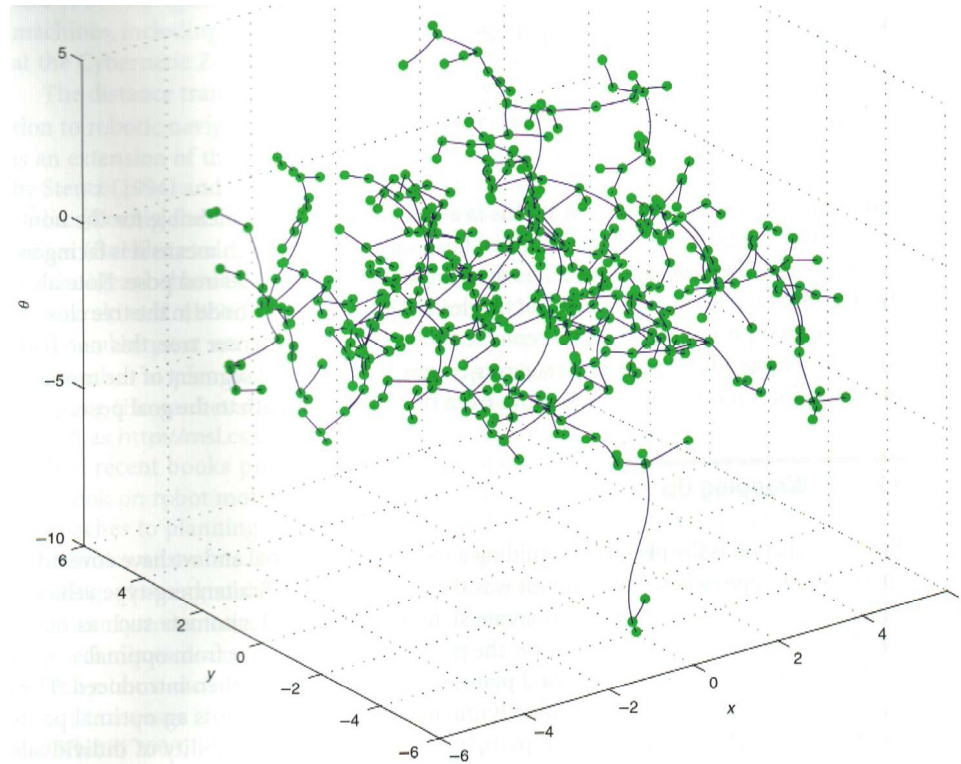


Corke, *Robotics, Vision and Control. Fundamental Algorithms in Matlab*

2.-Navegación deliberativa

Generación de rutas

- RRT, Rapidly-exploring Random Tree



Corke, *Robotics, Vision and Control. Fundamental Algorithms in Matlab*

2.-Navegación deliberativa

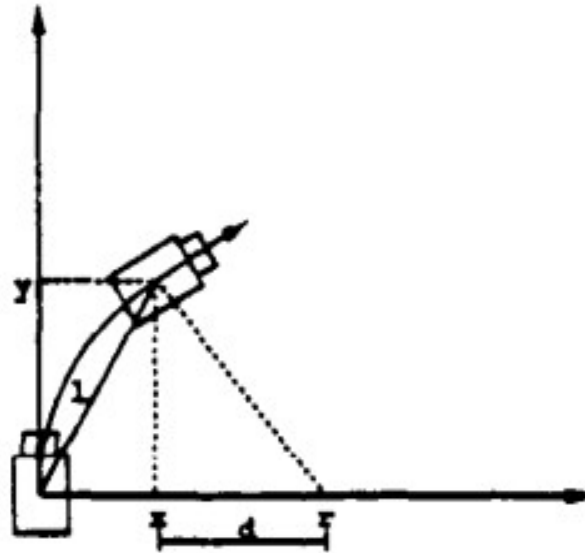
Técnicas de seguimiento

- Técnica muy conocida: **Pure Pursuit o Persecución Pura.**
 - Artículos interesantes:
 - Amidi O. (1990), *Integrated Mobile Robot Control*, CMU-RI-TR-90-177
 - Coulter R.C. (1992), *Implementation of the Pure Pursuit Path Tracking Algorithm*, CMU-RI-TR-92-01.
 - **Objetivo: calcular la curvatura que llevará al robot desde su posición a otra posición dada que se encuentra a una cierta distancia hacia delante.**
 - Esa distancia se denomina distancia *lookahead*.
 - **Conforme el vehículo avanza se van escogiendo nuevos puntos objetivos.**
-

2.-Navegación deliberativa

Técnicas de seguimiento

$$\begin{aligned}x + d &= r \rightarrow d = r - x \\d^2 + y^2 &= r^2 \rightarrow (r - x)^2 + y^2 = r^2\end{aligned}$$



$$r = \frac{x^2 + y^2}{2x}$$

$$\gamma = \left(\frac{2}{r^2}\right)x$$

2.-Navegación deliberativa

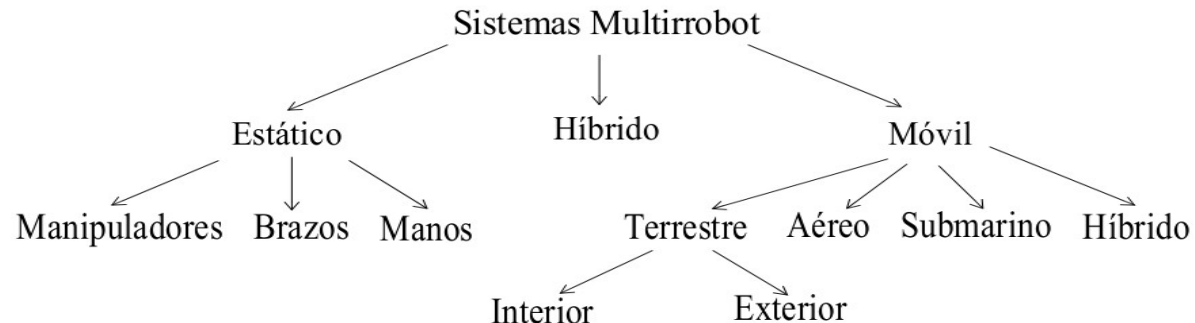
Técnicas de seguimiento

- La curvatura se relaciona con el desplazamiento en el eje x del punto objetivo, y la distancia lookahead.
- Así el pure-pursuit se asemeja a un controlador proporcional:
 - El error es el desplazamiento en el eje x del punto objetivo.
 - La ganancia K_p es $2/l^2$.
- A mayor lookahead (objetivo más alejado) K_p es menor, por lo que el objetivo se alcanzará suavemente y con más error; a menor lookahead, el objetivo se alcanza más precisamente pero con oscilaciones.

2.-Navegación deliberativa

Navegación multirrobot

- Sistema multirrobot: conjunto de robots en un entorno común.
- Caracterizable por diferentes aspectos:
 - Tipo de robot
 - Interacción entre elementos del sistema



2.-Navegación deliberativa

Navegación multirrobot

- Tipos de interacción: al compartir espacio surgen relaciones entre ellos, es decir, un comportamiento colectivo.
 - Apatía o indiferencia: la tarea de cada robot es independiente de los demás, y no hay beneficio en la colaboración.
 - Cooperación: los robots colaboran para lograr un fin común. Ejemplos: recolección o *foraging*, apacentamiento o *grazing*.
 - Competición o antagonismo: los objetivos de los robots son incompatibles entre sí. Ejemplos: persecución-evasión (*predator-prey*), fútbol robótico.

2.-Navegación deliberativa

Navegación multirrobot

- Ventajas de los sistemas multirrobot: dependen de la tarea y el entorno (a veces un sistema multirrobot no es útil):
 - Tareas con un ámbito geográfico extenso con una cierta sincronización.
 - División del trabajo.
 - Costes: puede resultar más rentable un conjunto de robots pequeños, baratos y versátiles frente a un único robot costoso.
 - Análisis de modelos sociales, biológicos o distribuidos.

Tareas típicas: mantenimiento de formación, aplicaciones militares (desactivación de minas), transporte-exploración-reconocimiento de entornos...

2.-Navegación deliberativa

Navegación multirrobot

- Problema de la navegación multirrobot: los restantes robots son obstáculos móviles.
- Por tanto, las técnicas de navegación en sistemas multirrobot deben considerar este hecho a la hora de definir trayectorias para los distintos robots, garantizando que el movimiento de los mismos sea seguro.

2.-Navegación deliberativa

Navegación multirrobot

- Clasificación de soluciones basada en dos criterios:
 - Con o sin perfil de velocidad
 - Planificación centralizada o desacoplada

Nivel	Eje	Valor	
I	Definición del perfil de velocidad	<i>Sin perfil de velocidad</i>	
		<i>Trayectoria</i>	
II	Tipo de planificación	<i>Centralizada</i>	
		<i>Desacoplada</i>	<i>Priorizada</i>
			<i>Coordinación de caminos</i>
			<i>Presencia de obstáculos móviles</i>

2.-Navegación deliberativa

Navegación multirrobot

- Criterio del perfil de velocidad:
 - Sin perfil de velocidad: suponemos que el robot se mueve a una velocidad constante y suficientemente baja para no causar problemas.
 - Con perfil de velocidad: sí se genera una trayectoria en la que se consideran las posibles restricciones de velocidad a la que se ve sometido el vehículo.

2.-Navegación deliberativa

Navegación multirrobot

- Criterio del tipo de planificación (admite algunas variaciones):
 - Latombe J.C., (1991), *Robot Motion Planning*, Kluwer Academic Publishers.
 - **Centralizada**: calcula todos los caminos simultáneamente, como si se tratara de un único robot compuesto. Es completa (si hay una solución, la encuentra), pero computacionalmente costosa.
 - **Desacoplada**: se calculan los caminos de cada robot individualmente, y después se analizan las posibles interacciones entre ellos. Es menos costosa, pero no es completa, y puede perder soluciones.

2.-Navegación deliberativa

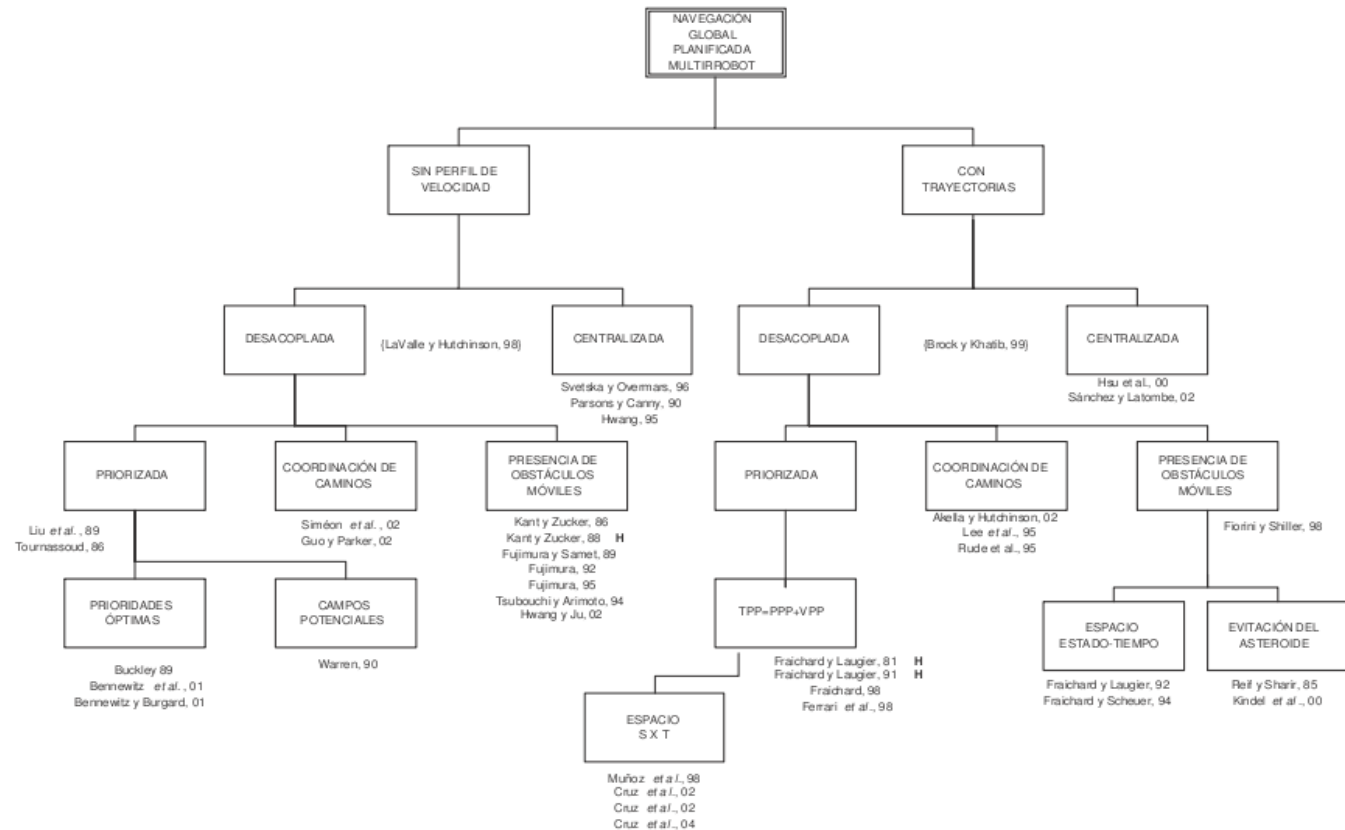
Navegación multirrobot

- Planificación desacoplada: hay varias metodologías
 - Priorizadas: los caminos se calculan de uno en uno, tomando aquellos robots cuyo perfil de velocidad ya se ha calculado como obstáculos móviles para los restantes. El orden de selección de los robots se basa en algún tipo de priorización. Se plantea en el espacio de configuraciones espacio-tiempo, obtenido al añadir la dimensión temporal al espacio de configuraciones tradicional. ¡El tiempo avanza, no retrocede!
 - Coordinación de caminos: utiliza métodos de planificación o scheduling en los que el espacio es el recurso compartido. No deben producirse colisiones ni bloqueos.
 - Presencia de obstáculos móviles: un único robot en un entorno dinámico con presencia de obstáculos móviles.

2.-Navegación deliberativa

Navegación multirrobot

- Clasificación final de soluciones



2.-Navegación deliberativa

- Ejercicio: movimiento hacia un punto

Usando el modelo de la bicicleta ya conocido como subsistema, implantar un modelo de un vehículo que tenga como objetivo el punto del plano $(x^*, y^*) = (5, 5)$.

El resto de especificaciones del problema son las siguientes:

- Como posición inicial se pueden usar las tuplas: $x_0 = [8, 5, \pi/2]$, $x_0 = [5, 8, \pi]$ y $x_0 = [2, 5, -\pi/2]$.
- La velocidad de entrada al modelo de la bicicleta será un controlador proporcional a la distancia euclídea al objetivo, con constante $K_v = 0.5$
- Para el ángulo de guiado vamos a usar un controlador proporcional sobre la diferencia angular entre la orientación final del vehículo θ^* y la orientación θ suministrada por el modelo de la bicicleta. La constante del controlador será $K_h = 4$.
- La diferencia angular calcula la diferencia de dos ángulos, devolviendo siempre un resultado en el intervalo $[-\pi, \pi]$.
- La orientación final del vehículo en el objetivo se calcula como: $\theta^* = \arctan\left(\frac{y^* - y}{x^* - x}\right)$

Robots móviles: navegación

- 1.- Navegación de robots móviles
- 2.- Navegación deliberativa
- 3.- Navegación reactiva

3.-Navegación reactiva

Conceptos

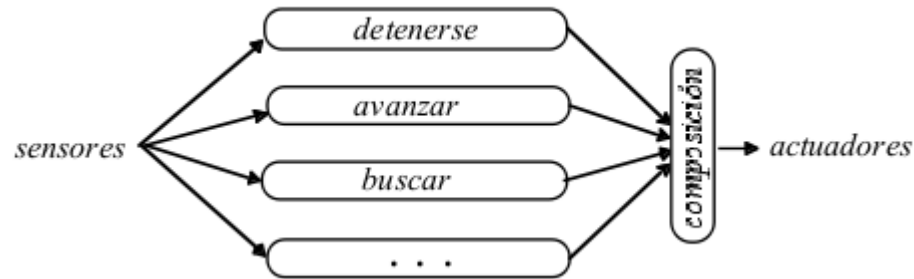
- Definición
- Ventajas y desventajas
- Vehículos de Grey Walter y Braitenberg
- Campos potenciales
- Algoritmo del bicho/Bug

Soo

3.-Navegación reactiva

Definición

- En la navegación reactiva, el robot reacciona directamente ante los estímulos, sin planificación previa.



- Basada en conceptos biológicos muchas veces: organismos simples (hormigas, abejas) son capaces de navegaciones complejas (<http://cyberneticzoo.com/>)

3.-Navegación reactiva

Ventajas y desventajas

- Ventajas

- Adaptación a cambios en el entorno.
- Son técnicas computacionalmente más ligeras

- Desventajas

- No tienen memoria, pueden perder el objetivo

3.-Navegación reactiva

Vehículos de Braitenberg/Grey Walter

- Son autómatas/robots que conectan sensores y actuadores directamente para producir comportamientos orientados a un objetivo.
 - Ejemplo de movimiento logrado: moverse hacia la luz (fototaxia/fototaxia negativa o fotostock), es decir, responder a un gradiente de luz.
-

3.-Navegación reactiva

Vehículos de Braitenberg/Grey Walter

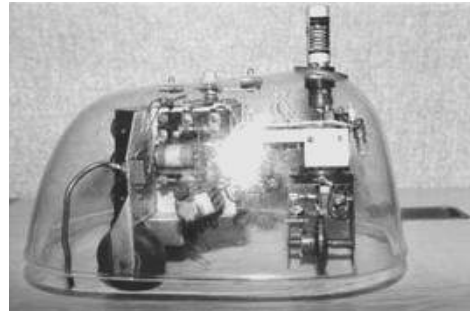
- Tortugas de Grey Walter:

- William Grey Walter (1910-1977), americano, estudios y vida profesional en UK
- Neurofisiólogo y robótico
- Trabajo robótico inspirado en neurología: cómo un pequeño número de células puede producir comportamientos complejos:
 - Robots (Machina Speculatrix): “Tortugas” (caparazón, lentas) Elmer y Elsie (1948-1949)
 - Movimiento hacia la luz y a la estación de recarga
 - Circuitos analógicos

3.-Navegación reactiva

Vehículos de Braitenberg/Grey Walter

- Tortugas de Grey Walter:



3.-Navegación reactiva

Vehículos de Braitenberg/Grey Walter

- Vehículos de Braitenberg:

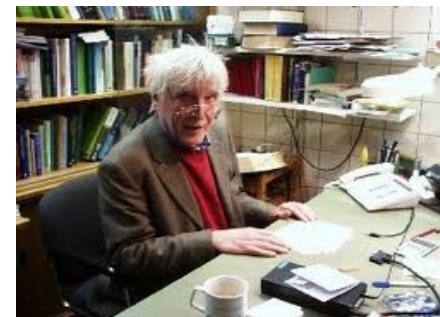
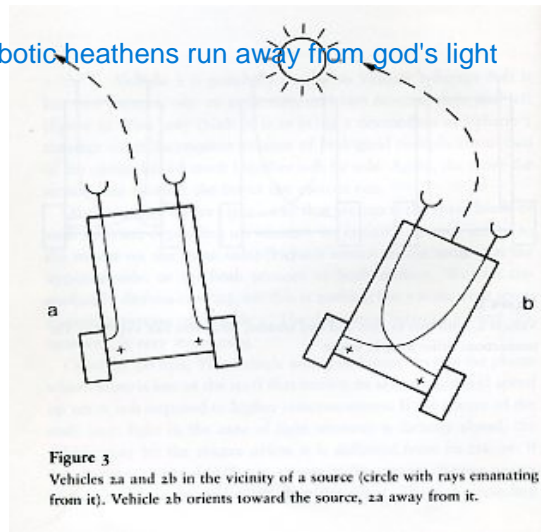
- Valentino Braitenberg (1926-2011), italo/austriaco
- Neurocientífico y robótico
- Vehículos de Braitenberg:
 - Colección de vehículos
 - Sensores y actuadores directamente conectados
 - Dependiendo de la conexión el robot logra diferentes comportamientos: “miedo”, “amor”...

3.-Navegación reactiva

Vehículos de Braitenberg/Grey Walter

- Vehículos de Braitenberg:

The god forsaken robotic heathens run away from god's light



Vehicles : experiments in synthetic psychology / Valentino Braitenberg

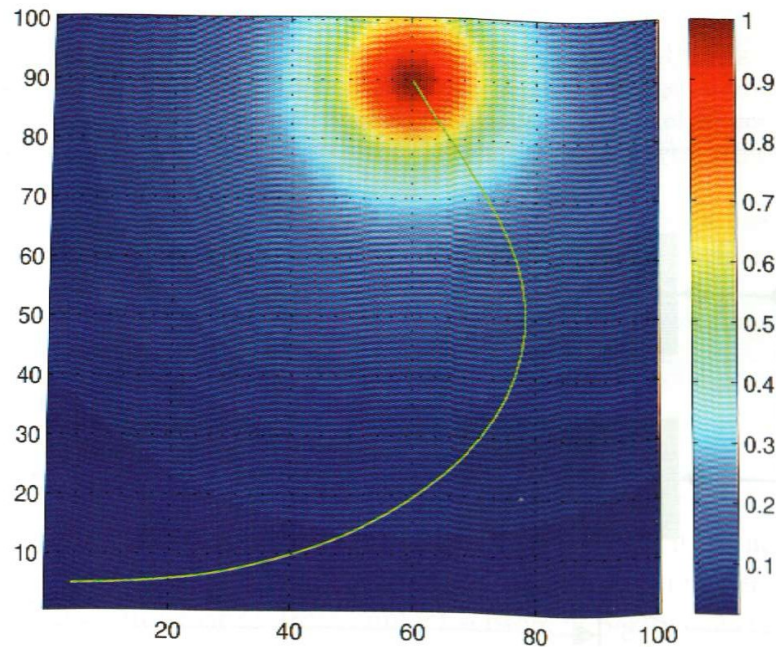
Disponible en Jábega

3.-Navegación reactiva

Vehículos de Braitenberg/Grey Walter

- Vehículos de Braitenberg:

En verdad es al revés lo que dicen es que va al sol que más calienta



Corke, *Robotics, Vision and Control. Fundamental Algorithms in Matlab*

3.-Navegación reactiva

Vehículos de Braitenberg/Grey Walter

- Vehículos de Braitenberg:

- Hemos logrado un comportamiento: ir hacia la luz
 - ¿Pueden añadirse más? Sí, por ejemplo, evitación de obstáculos...
 - Resultado final: comportamiento emergente (de nuevo, base biológica: hormigas, abejas)
 - Problema de la fusión comportamientos: ¿todos tienen el mismo peso? ¿Cómo se organizan?

3.-Navegación reactiva

Campos potenciales

Es un poco como trabajar con imanes, el objetivo nos atrae y el obstáculo nos repele.

- Suponemos que el robot se mueve en un campo potencial artificial.
- **Objetivo final: potencial atractivo. Debe disminuir con la distancia.**
- **Obstáculos: potencial repulsivo. Debe afectar sólo en las proximidades del obstáculo.**
- **Iterativamente, en cada momento, se genera una fuerza resultado de sumar la fuerza de atracción al objetivo y la fuerza de repulsión a los obstáculos. A partir de esta fuerza, se obtienen las consignas de navegación. En el destino, la fuerza debe ser nula.**
- **Problema: aparición de mínimos locales, con lo que no se alcanza el objetivo final.**

3.-Navegación reactiva

Algoritmo del Bicho/Bug

- Buscan un objetivo en presencia de obstáculos, pero sí incluyen algún tipo de lógica entre sensor y motor (máquina de estados, memoria...)
- Hay varios algoritmos del bicho
- Ejemplo:
 - El robot sabe su posición inicial y su objetivo
 - Se mueve en una rejilla, ocupa una celda, no hay restricciones holonómicas y por tanto puede moverse a cualquier celda.
 - Sólo conoce su entorno local y el objetivo

3.-Navegación reactiva

Algoritmo del Bicho/Bug

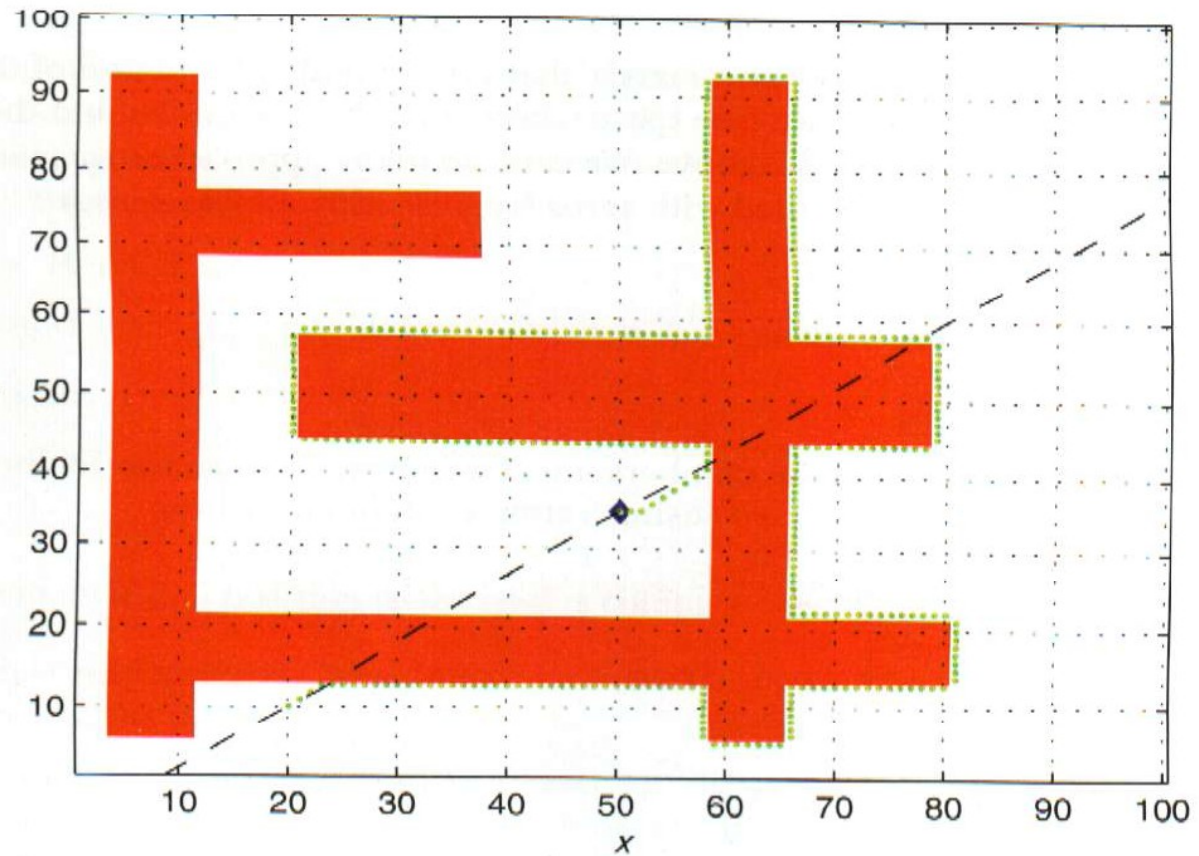
- Algoritmo:

- Trazar línea recta desde el inicio hasta el objetivo.
- Se sigue esa línea.
- Si no se puede por la presencia de un obstáculo, se sorteá el mismo siempre en el mismo sentido (antihorario) hasta que se encuentre un punto en esa línea más cercano al objetivo que el primero que se tuvo que abandonar.

- No es óptimo: se encuentra un camino, pero no es el mejor camino, ya que al no haber mapa perdemos la perspectiva global y tomamos caminos óptimos localmente.

3.-Navegación reactiva

Algoritmo del Bicho/Bug



Corke, *Robotics, Vision and Control. Fundamental Algorithms in Matlab*

3.-Navegación reactiva

Algoritmo del Bicho/Bug

- Ejercicio: navegación reactiva

Usando el modelo de la bicicleta ya conocido como subsistema, implantar un modelo de un vehículo de Braitenberg con un comportamiento de fototaxia. Supondremos que nuestro vehículo tendrá dos sensores en los lados del vehículo, uno al lado izquierdo y otro al lado derecho, a una distancia de 2 unidades en la dirección lateral del vehículo (eje y).

- El tiempo de simulación será de 200 segundos
- La postura de partida es [5,5,0].
- La fuente de luz se encuentra en la posición [60,90].
- Si la medida proporcionada por el sensor derecho es S_d , y la proporcionada por el sensor izquierdo es S_i , entonces:
 - La velocidad a suministrar al modelo de la bicicleta es $v=2-S_d-S_i$.
 - El ángulo de guiado es $=k(S_i-S_d)$, con $k = 5$.

Habrà que modelar cada sensor teniendo en cuenta su desplazamiento (traslación, rotación) respecto al vehículo. El gradiente que se percibe está definido en el intervalo [0,1] u viene dado por:

$$\text{Medida} = 200 / ((x - x_{\text{fuente}})^2 + (y - y_{\text{fuente}})^2 + 200);$$