

# Robots Móviles

Robótica reactiva. Comportamientos

Otto Colomina Pardo

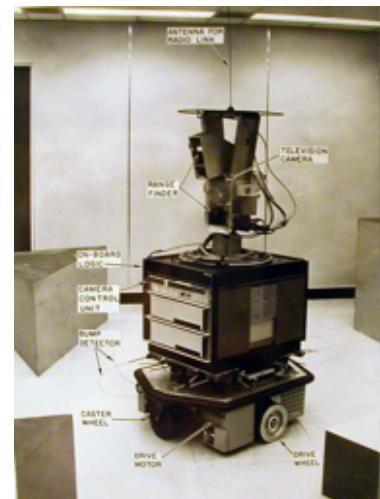
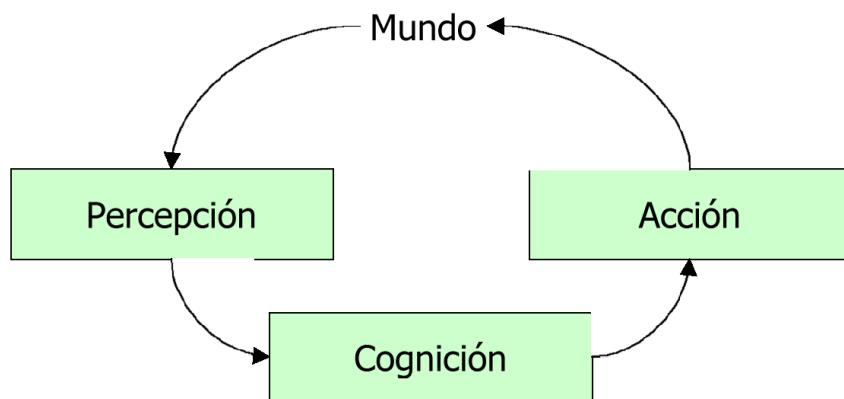
[otto@dccia.ua.es](mailto:otto@dccia.ua.es)

# Índice

Introducción a la Robótica  
reactiva  
Campos de potencial  
Subsumpción  
Secuenciación de conductas

# Robótica “tradicional”

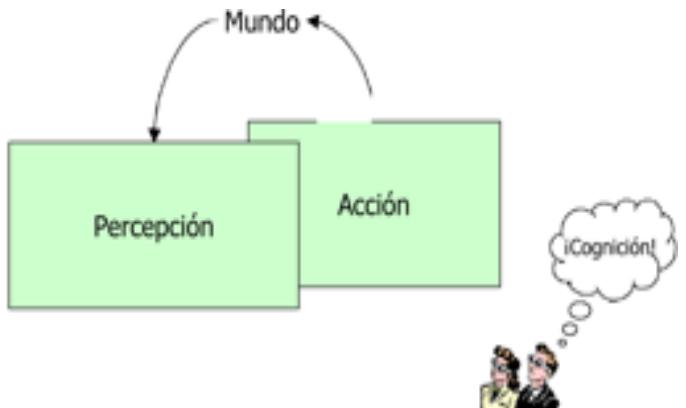
- Problemas:
  - Obliga a mantener un modelo complejo del mundo y de los objetivos del robot
  - Coste temporal de razonamiento/planificación



Robot “shakey” (1966-1972)

# Sistemas robóticos reactivos

- Son sistemas que acoplan la percepción a la acción, sin la intervención de representaciones abstractas o memoria histórica



## Elephants Don't Play Chess

Rodney A. Brooks

MIT Artificial Intelligence Laboratory, Cambridge, MA 02139, USA

Robotics and Autonomous Systems 6 (1990) 3-15

*Keywords:* Situated activity; Mobile robots; Planning; Subsumption architecture; Artificial Intelligence.



Rodney A. Brooks was born in Adelaide, Australia. He studied Mathematics at the Flinders University of South Australia and received a Ph.D. from Stanford in Computer Science in 1981. Since then he has held research associate positions at Carnegie Mellon University and the Massachusetts Institute of Technology and faculty positions at Stanford and M.I.T. He is currently an Associate Professor of Electrical Engineering and Computer Science at M.I.T. and a member of the Artificial Intelligence Laboratory where he leads the mobile robot group. He has authored two books, numerous scientific papers, and is the editor of the *International Journal of Computer Vision*.

Elephants don't play chess

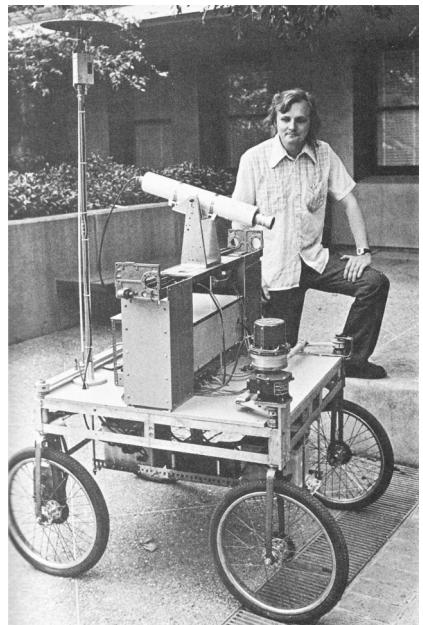
# ¿Por qué surgió la robótica reactiva?

## La historia del “carrito de Stanford”

Stanford Cart - 1966-1979



Hans Moravec 1977



# El enfoque “clásico”, aplicado al carrito

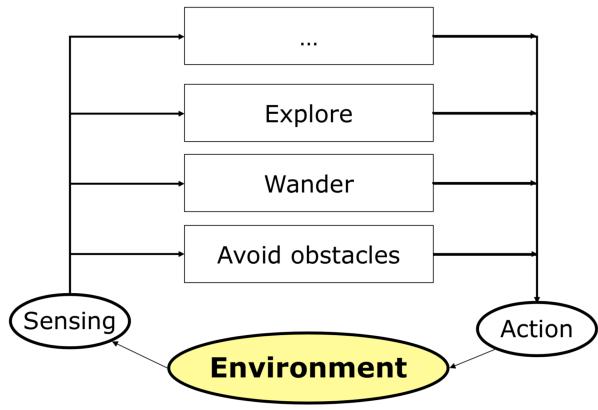
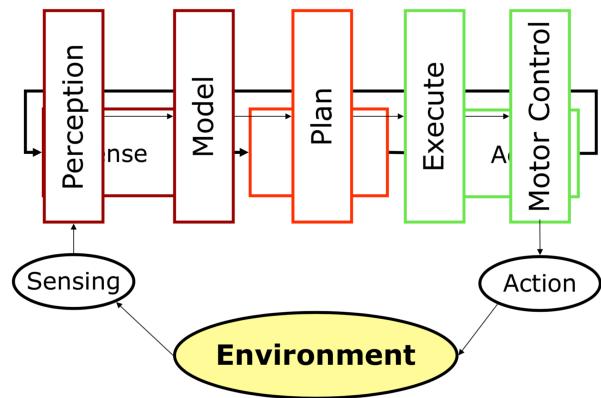
1. Tomar varias imágenes del entorno desde posiciones ligeramente distintas,
2. buscar puntos clave en una de ellas e identificarlos en las otras. Integrar esta información en un modelo 3D global del mundo
3. A partir de estos datos deducir cuánto se ha movido el carro
4. Calcular la dirección del movimiento según posición actual, objetivo y entorno
5. Moverse aproximadamente 1 m en la dirección deseada
6. Volver al paso 1

10-15  
minutos

En 1979 el carro cruzó de forma autónoma una habitación llena de obstáculos...  
**en 5 horas**



# Descomposición horizontal vs vertical



# Una mirada a la biología

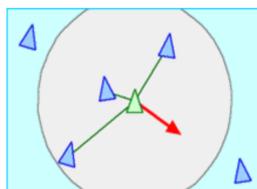
- **Psicología conductista:**
  - La acción viene determinada por los estímulos (perro de Pavlov)
- **Etología: estudio del comportamiento animal**
  - Los animales dan respuestas inteligentes e inmediatas al entorno (hormigas, abejas)
  - Conductas complejas surgen de combinar conductas simples

<https://www.youtube.com/watch?v=Cr4xc79tSYQ>

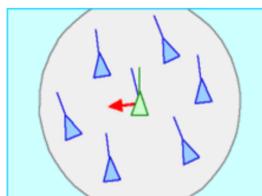
# Ejemplo: *boids* (Reynolds, 1986)

- Bandadas: comportamiento complejo que se puede conseguir combinando reglas simples que solo usan información local

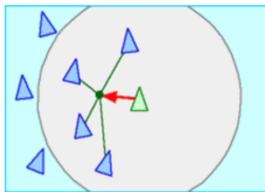
1. Separación: no acercarse demasiado a los vecinos

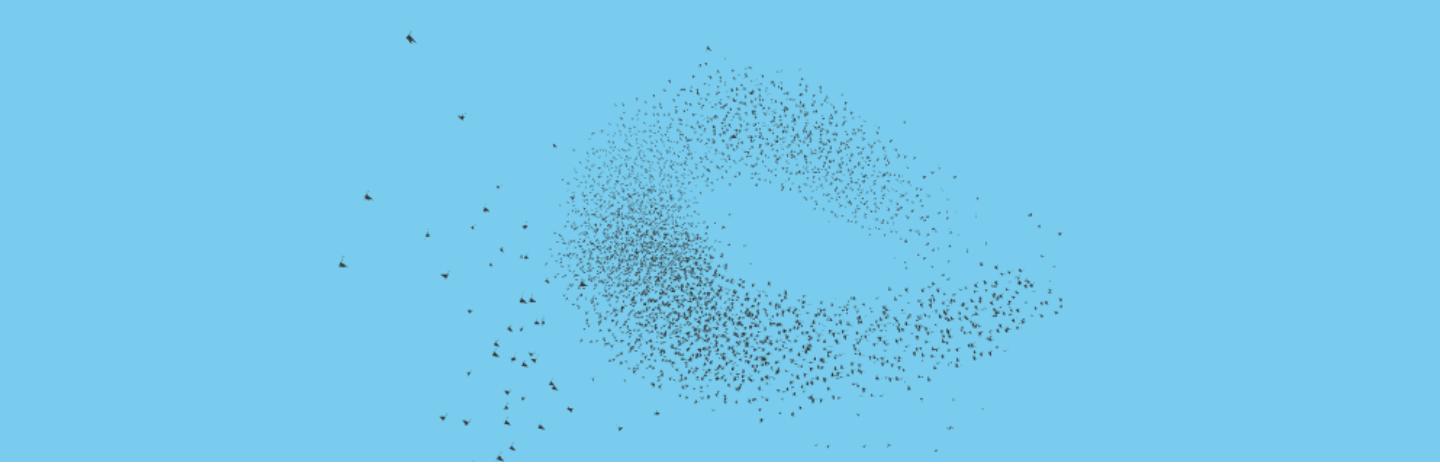


2. Alineación : viajar en la misma dirección que los vecinos



3. Cohesión : ir hacia el centro de los vecinos





<https://github.com/techcentaur/Flocking-Simulation>  
(Demo 3D en Javascript con WebGL)

La idea e implementación original es de Craig Reynolds (1986). Mantiene [una página](#) con información sobre el tema, aunque en la web podéis encontrar decenas de implementaciones, como la mostrada aquí



# Características de los sistemas reactivos

- Los robots son “agentes situados” que operan en un “nicho ecológico”
- Los comportamientos son los bloques básicos, y el comportamiento global es *emergente*
- Cada comportamiento tiene su propia información sensorial, no hay modelos globales
- Se suele citar el comportamiento animal como base para comportamientos particulares
- Como consecuencia de todo lo anterior, los sistemas son modulares

(Introduction to AI Robotics, R. Murphy, cap.4)

# Características de los sistemas reactivos

- Los robots son “agentes situados” que operan en un “nicho ecológico”
- Los comportamientos son los b... comportamiento global es emergente
- Cada comportamiento tiene su base sensorial, no hay modelos globales
- Se suele citar el comportamiento base para comportamientos parciales
- Como consecuencia de todo lo anterior, los sistemas son modulares

Los robots no están “fuera del mundo” sino integrados en él y “adaptados” al entorno

(Introduction to AI Robotics, R. Murphy, cap.4)

# Características de los sistemas reactivos

- Los robots son “agentes situados” que operan en un “nicho ecológico”
- **Los comportamientos son los bloques básicos, y el comportamiento global es emergente**
- Cada comportamiento es sensorial, reactivo
- Se suele combinarlos en base para formar un sistema
- Como combinación de sistemas se obtiene el comportamiento global

**El comportamiento global surge de la combinación de todos los comportamientos. Las diferentes arquitecturas reactivas se distinguen en cómo combinar esos comportamientos**

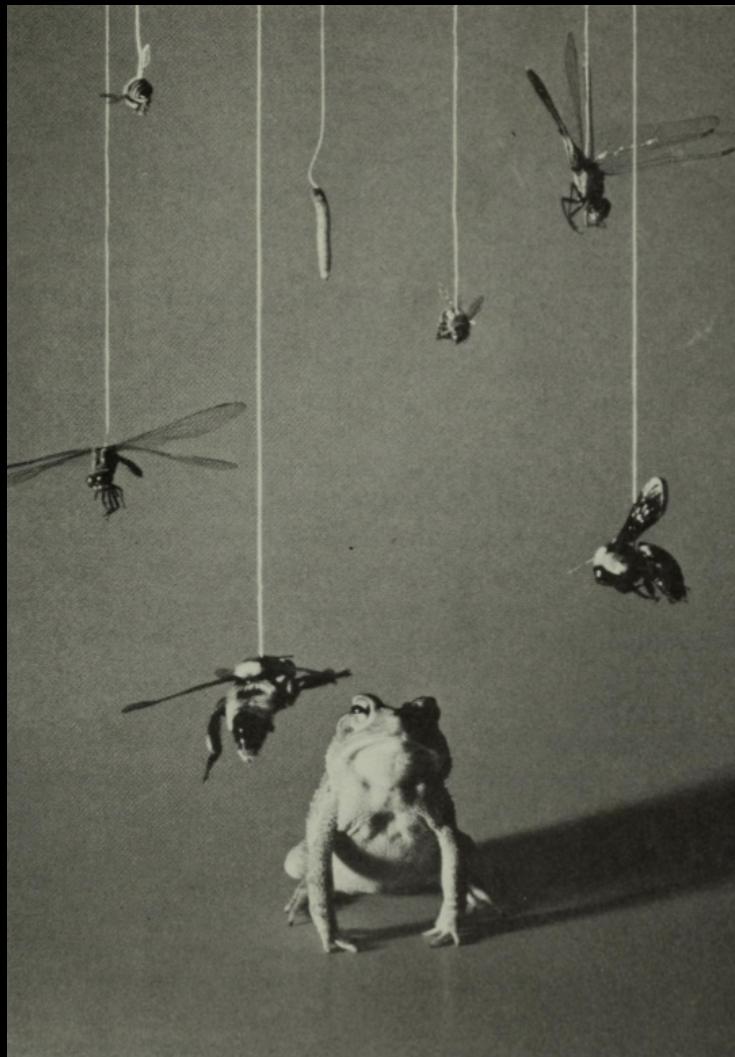
(Introduction to AI Robotics, R. Murphy, cap.4)

# Características de los sistemas reactivos

- Los robots son “agentes situados” que operan en un “nicho ecológico”
- Los comportamientos son los bloques básicos, y el comportamiento global es *emergente*
- **Cada comportamiento tiene su propia información sensorial, no hay modelos globales**
- Se suele citar el comportamiento animal como base para comportamientos particulares
- Como consecuencia de todo lo anterior, los sistemas son modulares  
(Introduction to AI Robotics, R. Murphy, cap.4)

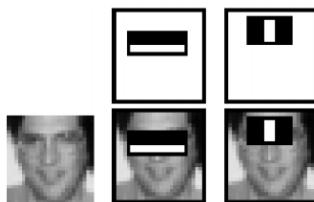
# Percepción en un *behavior*

- El sistema de coordenadas es **local**, relativo al robot, no hay un mapa global
- Cada comportamiento obtiene **solo la información que necesita**, no hay un análisis global del entorno
  - Detectar un color, una forma
  - Detectar el obstáculo más cercano
  - Detectar una pared

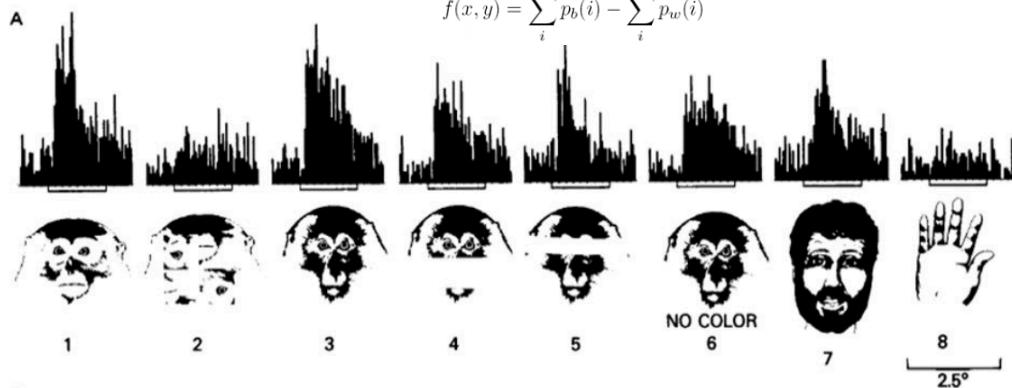


# Percepción en un *behavior*

- Que no haya un modelo global **no quiere decir que procesar la información sensorial sea trivial**, por ejemplo: detectar una cara



Detección de caras con el método clásico de Viola y Jones (Haar Features + AdaBoost)



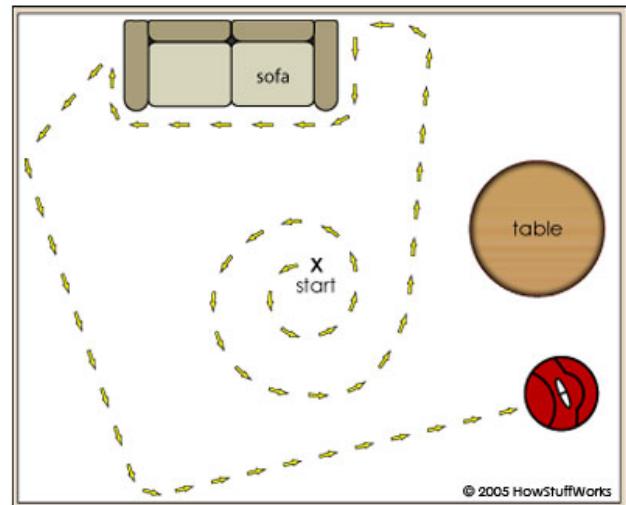
Neuronas detectoras de caras y manos en la corteza cerebral de los monos

# Características de los sistemas reactivos

- Los robots son “agentes situados” que operan en un “nicho ecológico”
- Los comportamientos son los bloques básicos, y el comportamiento global es *emergente*
- Cada comportamiento tiene su propia información sensorial, no hay modelos globales
- **Se suele citar el comportamiento animal como base para el diseño de comportamientos**
- Como consecuencia de todo lo anterior, los sistemas son modulares

(Introduction to AI Robotics, R. Murphy, cap.4)

# Inspiración biológica: iRobot Roomba



“

When it starts you'll notice a spiral pattern, it'll spiral out over a larger and larger area until it hits an object. When it finds an object, it will follow along the edge of that object for a period of time, and then it will start criss-crossing, trying to figure out the largest distance it can go without hitting another object, and that's helping it figure out how large the space is, but if it goes for too long a period of time without hitting a wall, it's going to start spiraling again, because it figures it's in a wide open space, and it's constantly calculating and figuring that out.

**The patterns that we chose and how the algorithm was originally developed was based off of behavior-based algorithms born out of MIT studying animals and how they go about searching areas for food.** When you look at how ants and bees go out and they search areas, these kinds of coverage and figuring all of that out comes from that research. It's not exact, obviously, I'm not saying we're honeybees, but it's that understanding of how to search out an area in nature that is the basis behind how our adaptive technology is developed.

”

[Entrevista con Nancy Dussault, iRobot](#)

# Características de los sistemas reactivos

- Los robots son “adaptados a su entorno”
- Un “nicho ecológico”
- Los comportamientos determinan las acciones
- Los comportamientos se basan en la información sensorial, no hay aprendizaje
- Cada comportamiento es un módulo simple
- Se suele citar el principio de modularidad como base para el diseño de comportamientos
- **Como consecuencia de todo lo anterior, los sistemas son modulares**

**Los comportamientos son como módulos del sistema, cohesionados pero relativamente independientes entre sí, se deberían poder desarrollar y probar por separado**

(Introduction to AI Robotics, R. Murphy, cap.4)

# Conductas elementales

- Una conducta viene definida por un **mapeado** entre los **sensores** (ej: sonar) y los **actuadores** (ej: motor)
- Ejemplo: **Seguir-pared:**
  - Sensores: distancia a los distintos sonares
  - Actuadores: modificación de la velocidad lineal y angular

# Implementación de conductas

**conducta ejemplo()**

*repetir {*

*p := percibir\_sensores();*

*s := procesar\_sensores(p);*

*r := calcular\_conducta(s);*

*Accionar\_actuadores(r);*

*} hasta fin\_conducta();*



Lecturas sonares



Ajustar recta a las  
lecturas. Calcular  
distancia a la recta, *dist*



Dirección: paralela a a la  
pared si *dist*>umbral,  
perpendicular en caso  
contrario  
velocidad: proporcional a  
*dist*

# Coordinación de conductas

- Podemos tener varias conductas activas a la vez en el sistema
  - Evitar obstáculo
  - Ir a objetivo
- Debemos coordinar la respuesta de las distintas conductas:
  - Métodos competitivos
  - Métodos cooperativos

# Dos “estilos clásicos” de robótica reactiva

- **Campos de potencial:** cada conducta es un “campo de fuerzas”. El comportamiento global surge de sumarlas todas



Ronald Arkin

- **Subsumpción:** cada conducta reside en una capa del sistema, que influye sobre las capas inferiores (modificando, inhibiendo, ...)



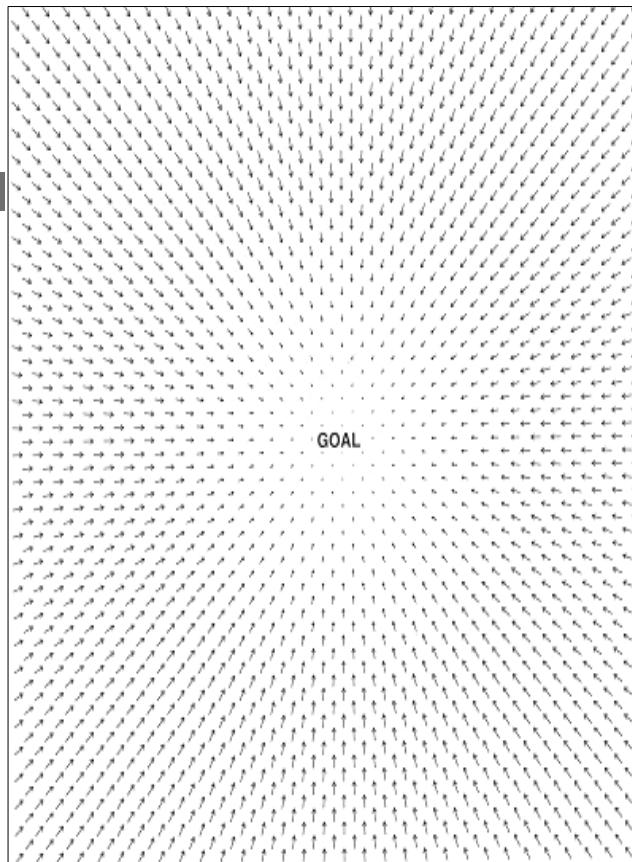
Rodney Brooks

# Índice

Introducción a la Robótica reactiva  
**Campos de potencial**  
Subsumpción  
Secuenciación de conductas

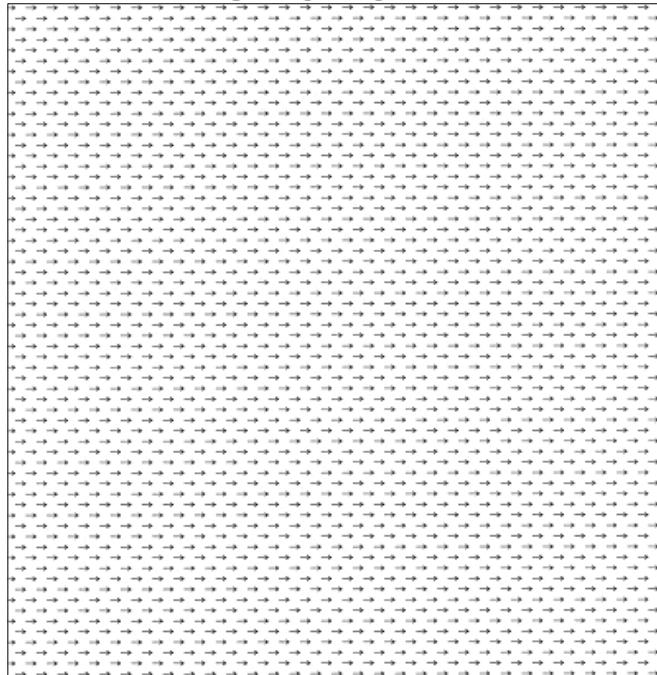
# Campos de potencial

- Las conductas se modelan como **campos de potencial** (“campos vectoriales” o “campos de fuerzas”), típicamente 2D
- **El campo no se calcula entero**, solo en el punto en que está el robot (**MUCHO más eficiente!**), aunque aquí mostraremos el campo entero para visualizarlo mejor

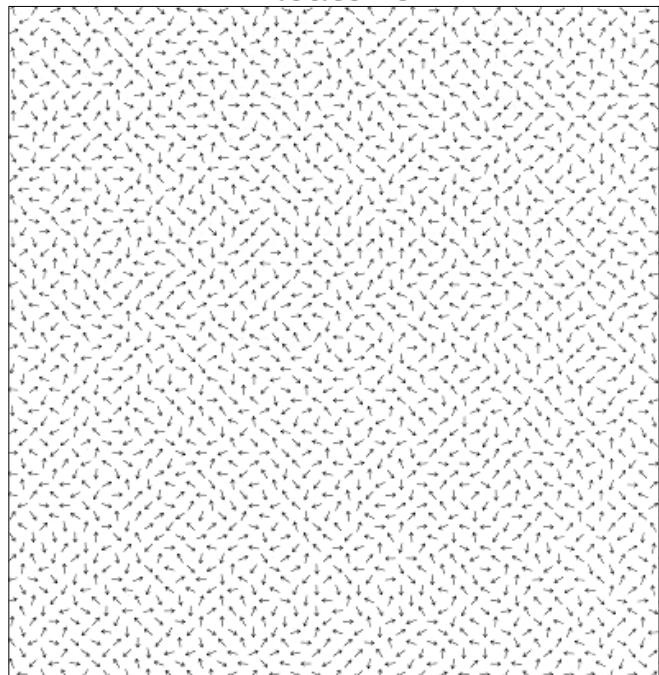


# Campos de potencial básicos (II)

Uniforme



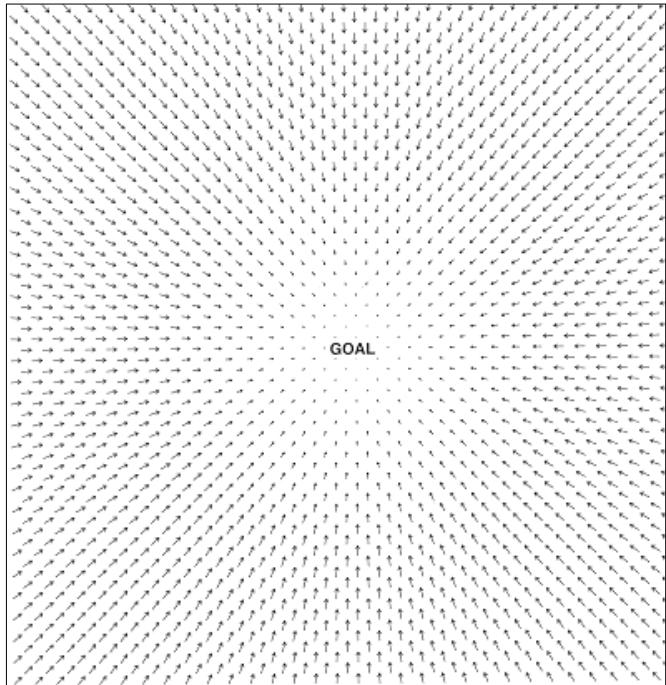
Aleatorio



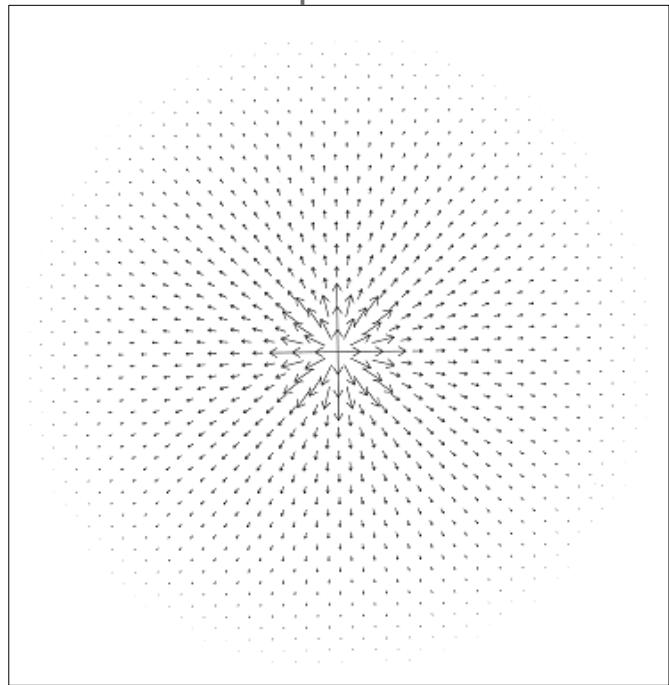
Según la fuente donde lo consultéis encontraréis unos u otros.  
Combinándolos obtendremos comportamientos complejos.

# Campos de potencial básicos (II)

Atractivo

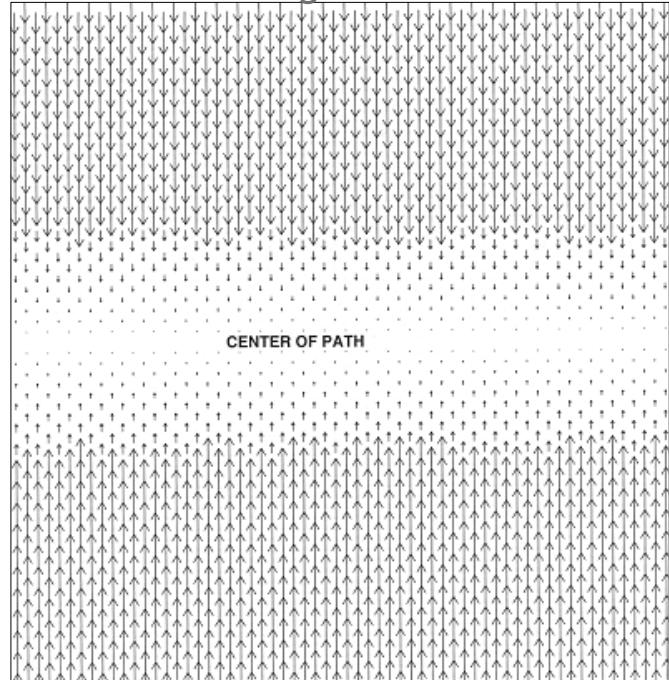


Repulsivo

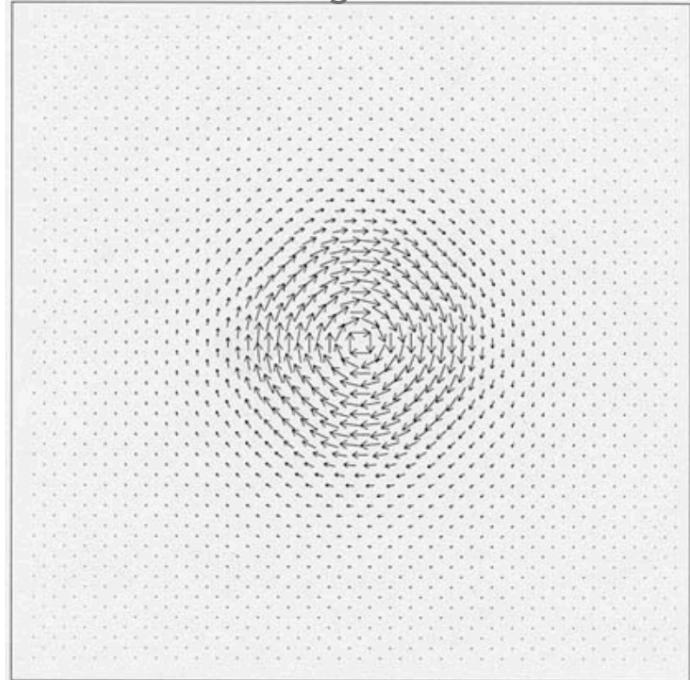


# Campos de potencial básicos (III)

Seguir camino

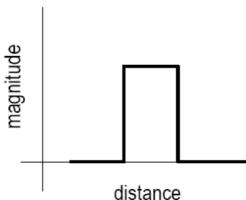


Tangencial

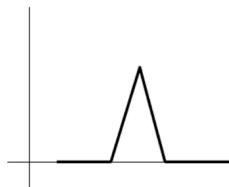


# Campos no uniformes

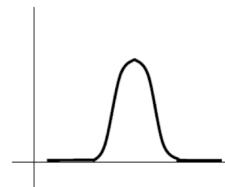
- Observar que en general los campos no son uniformes: por ejemplo:
  - En el campo repulsivo la magnitud es mayor cuanto más cerca del obstáculo
  - En el campo atractivo la magnitud es menor cuanto más cerca del objetivo
- Los campos uniformes llevarían a un comportamiento “brusco” justo en el borde del campo



Constant



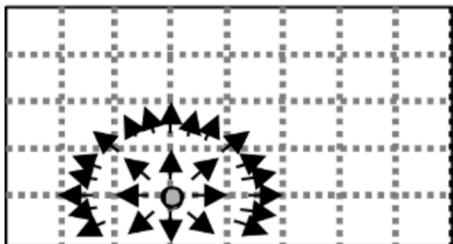
Linear Dropoff



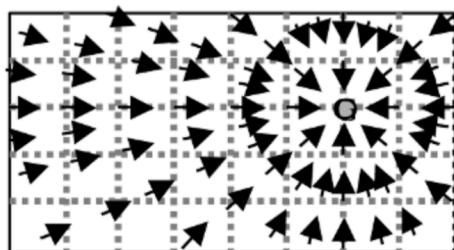
Exponential  
Dropoff

# Combinar conductas en campos de potencial

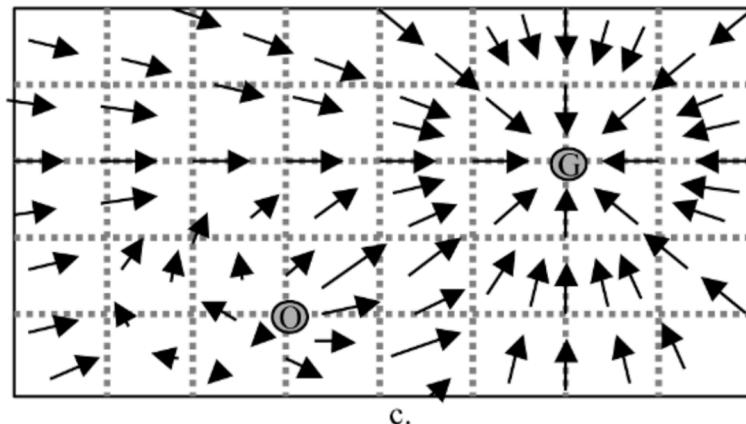
- Sumar los vectores de cada campo



a.



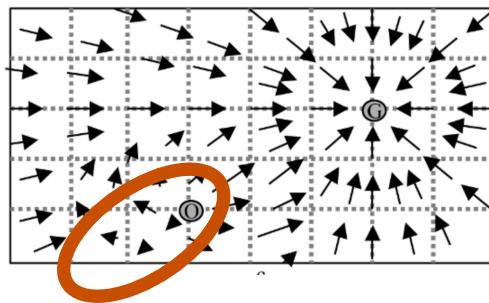
b.



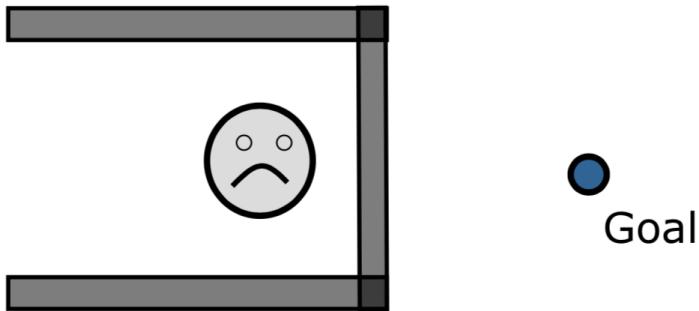
c.

# Problemas de los campos de potencial

- **Intervalo de actualización:** cuanto mayor es la frecuencia de actualización más “suave” es la trayectoria del robot
- **Robots no holonómicos:** la trayectoria generada por el campo supone que el robot puede seguir cualquier dirección de movimiento en un instante dado, lo que no es cierto
- **Mínimos locales:** situaciones en las que la suma de los campos es 0



# Cómo resolver los mínimos locales



- Campo ruido/aleatorio para intentar “empujar” al robot fuera del mínimo local
- Backtracking
- Planificación para “seguir el muro”
- Campo repulsivo para las áreas ya visitadas, mayor cuanto más tiempo haya estado el robot

# Índice

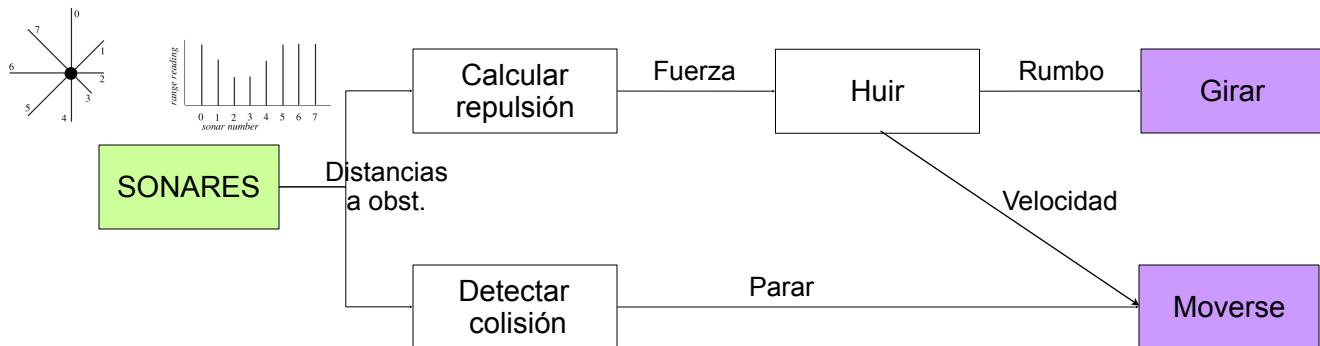
Introducción a la Robótica reactiva  
Campos de potencial  
**Subsumpción**  
Secuenciación de conductas

# Arquitectura *subsumption*

- Control por niveles
- Cada nivel utiliza elementos del nivel inferior
- Un nivel de competencia superior subsume algunas competencias del nivel inferior
- La adaptación de un sistema a otro supone una adaptación del sistema completo

# Ejemplo *subsumption*

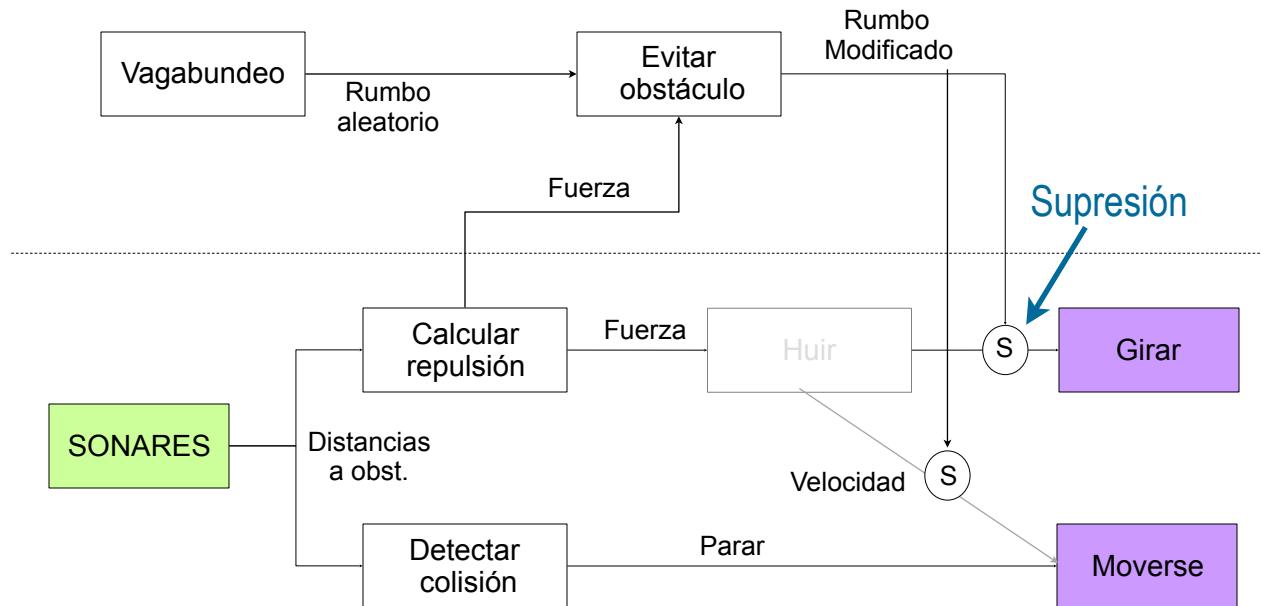
- Nivel 0: evitar obstáculos



El robot en principio no se mueve pero si se le acerca un obstáculo (p.ej. una persona) “huirá” de él

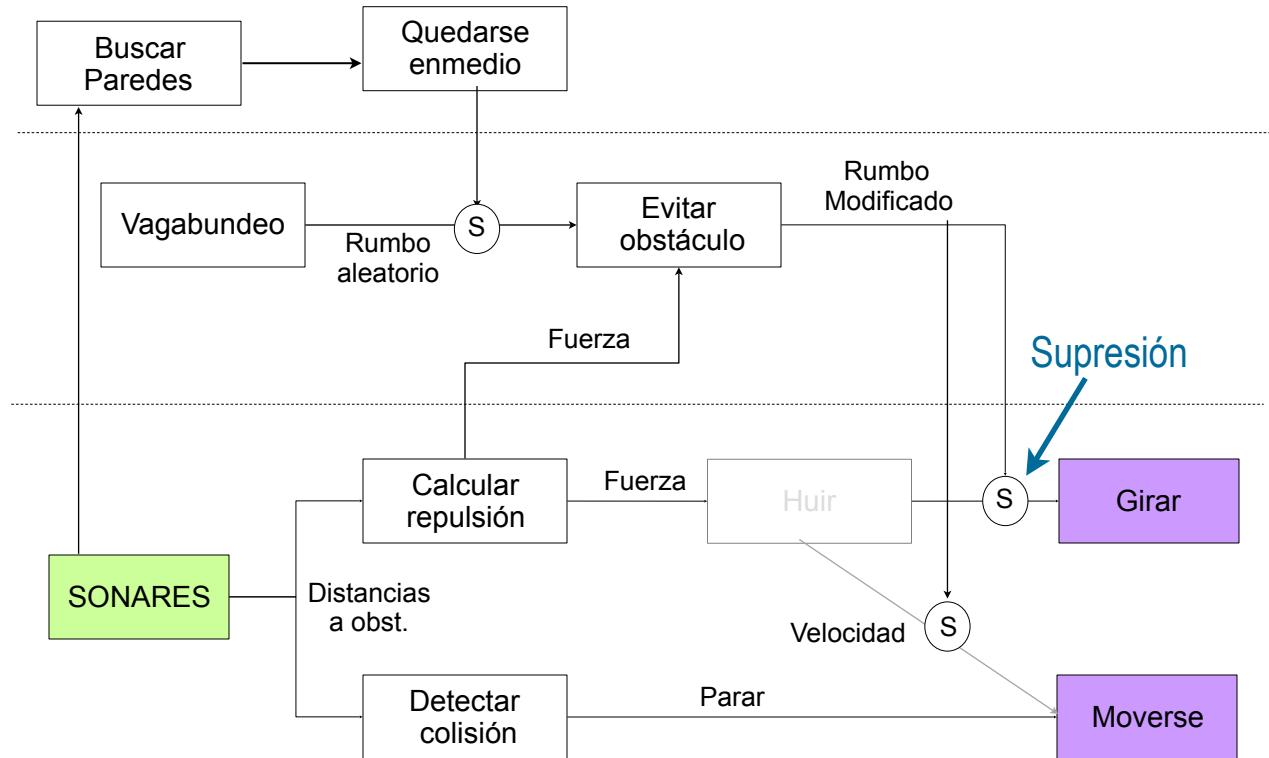
# Ejemplo *subsumption*

- Nivel 1: vagabundear



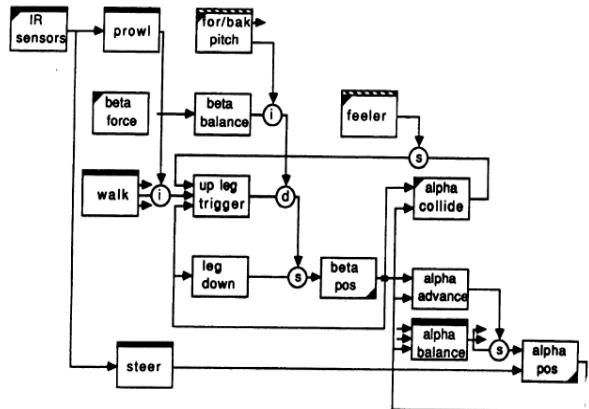
# Ejemplo *subsumption*

## Nivel 2: seguir pasillo



# Genghis

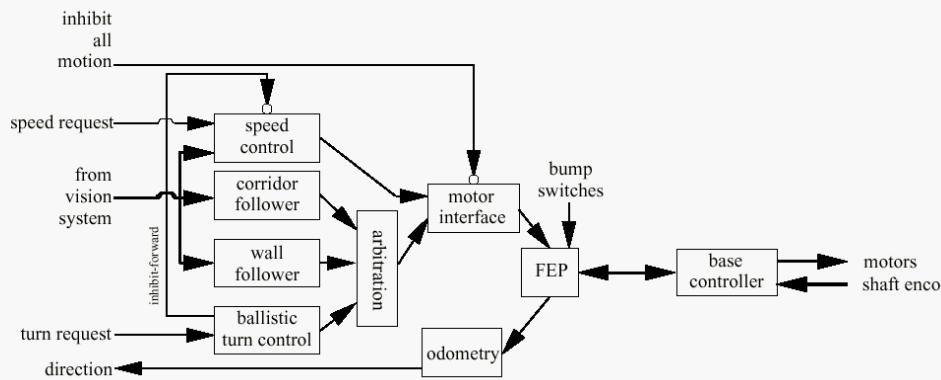
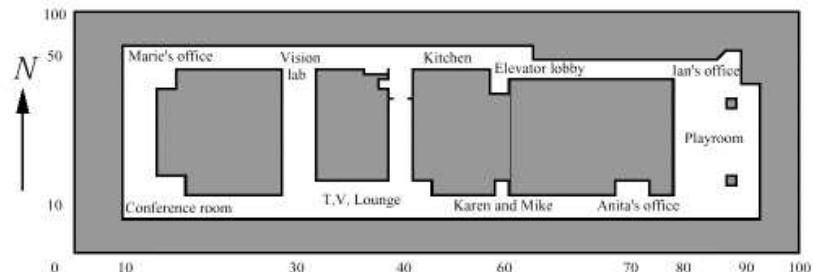
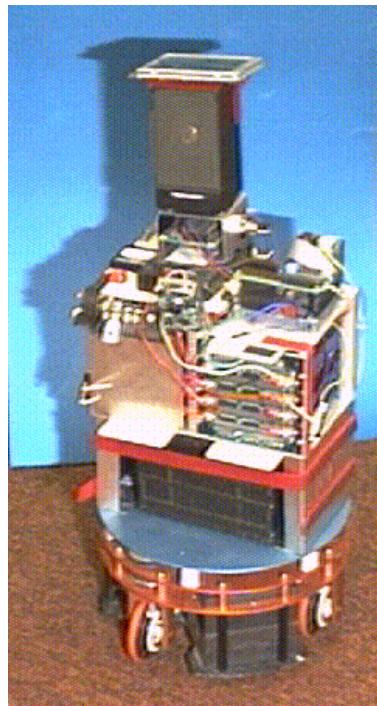
- Controlado mediante *subsumpción* por una red de 57 máquinas de estados finitos (cada *behavior* es una FSM)



<https://www.youtube.com/watch?v=1j6ClOWRng>

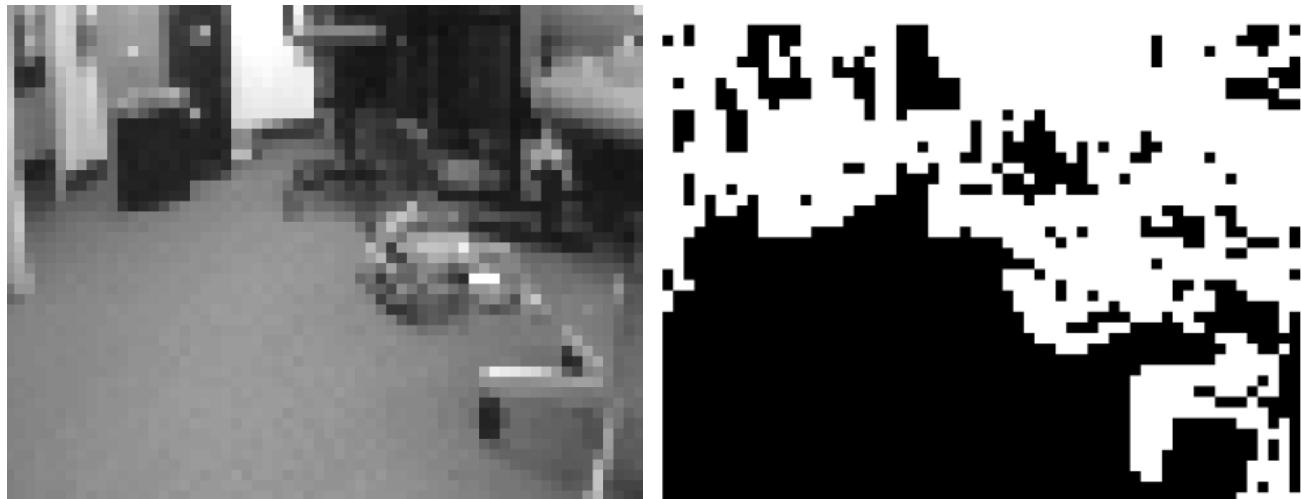
# Polly (1992-94) ¿Llevando la subsumpción a los límites?

## Guía robótico del MIT



de <http://www.cs.hmc.edu/courses/2008/spring/cs154/lectures/index.html> (Zachary Dodd)

# Sistema de visión de Polly

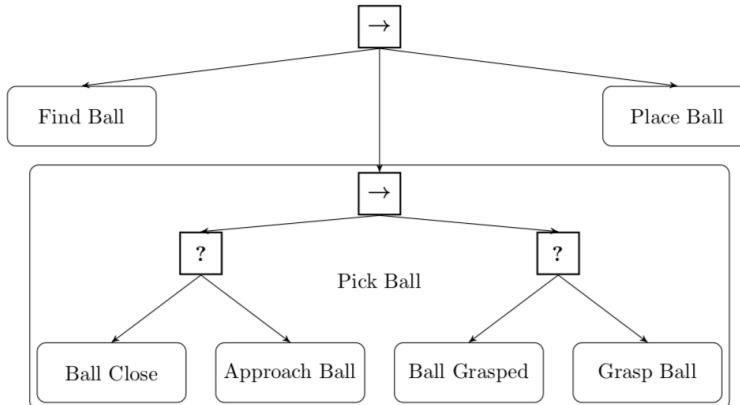
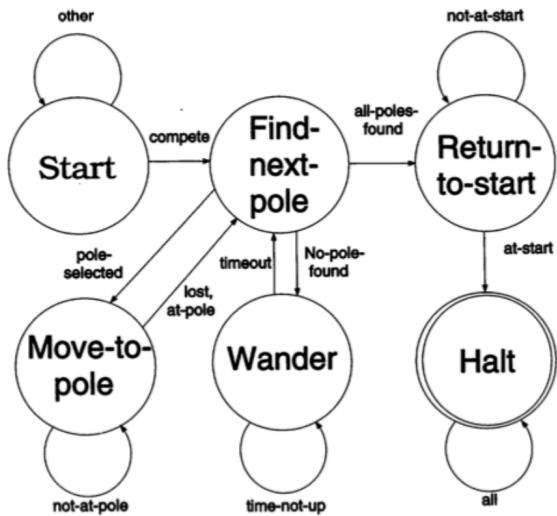


- El suelo es lo que está inmediatamente delante con una determinada textura (alfombra)
- Los obstáculos lo de color distinto al suelo
- Los pasillos tienen líneas que convergen

de <http://www.cs.hmc.edu/courses/2008/spring/cs154/lectures/index.html> (Zachary Dodd)

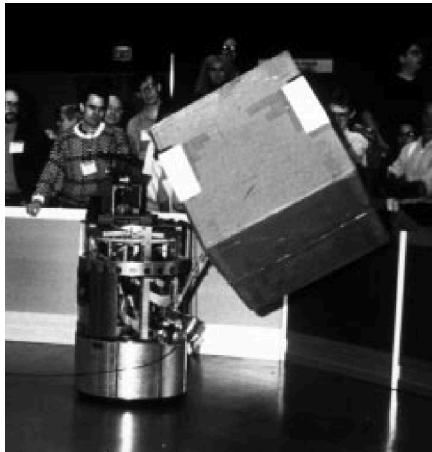
# Otras formas de combinar comportamientos

- Nos permiten secuenciarlos en un orden, alternarlos, etc.
  - Máquinas de estados finitos
  - Behavior trees



# Ejemplo: “Pick up the trash”

- Competición “clásica” de robots realizada en los 90 por la AAAI/IJCAI
- En un entorno en que hay varios tipos de basura (botes de refresco/vasos de papel) recogerla y llevarla a su sitio (cubo basura/reciclaje)



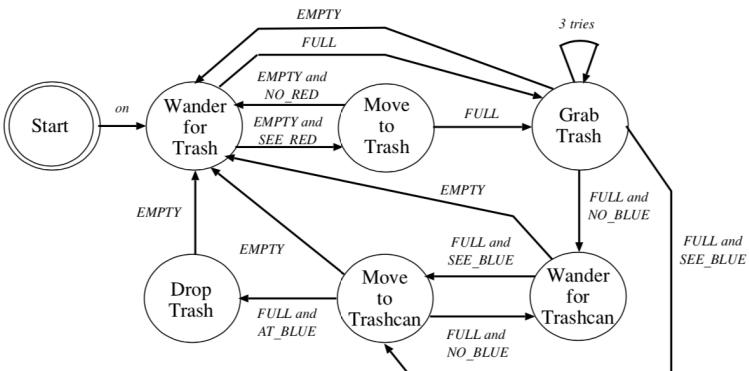
Xavier, de CMU, [1994](#)



[Lola](#), el robot ganador en 1995

# Máquina de estados finitos

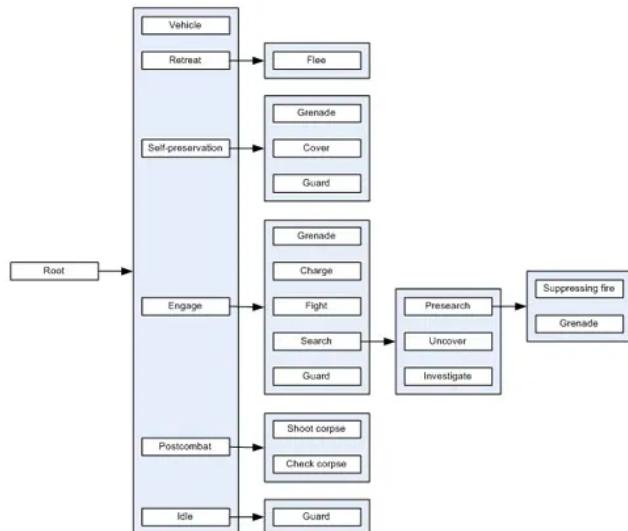
- Conjunto de: {estados ( $q$ ), entradas ( $\sigma$ ), transiciones ( $\delta(q, \sigma)$ ), estado inicial ( $s$ ), estados finales ( $f$ )}



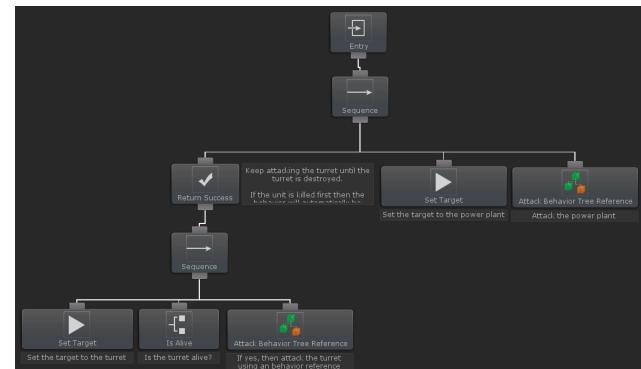
$q$	$\sigma$	$\delta(q, \sigma)$
start	on	wander for trash
wander for trash	EMPTY, SEE_RED	move to trash
wander for trash	FULL	grab trash
move to trash	FULL	grab trash
move to trash	EMPTY, NO_RED	wander for trash
grab trash	FULL, NO_BLUE	wander for trash can
grab trash	FULL, SEE_BLUE	move to trash can
grab trash	EMPTY	wander for trash
wander for trash can	EMPTY	wander for trash
wander for trash can	FULL, SEE_BLUE	move to trash can
move to trash can	EMPTY	wander for trash
move to trash can	FULL, AT_BLUE	drop trash
drop trash	EMPTY	wander for trash

# Behavior trees

- Alternativa a las máquinas de estados finitos para coordinar/secuenciar behaviors
- Muy usada en IA para videojuegos, no tan extendida en robótica



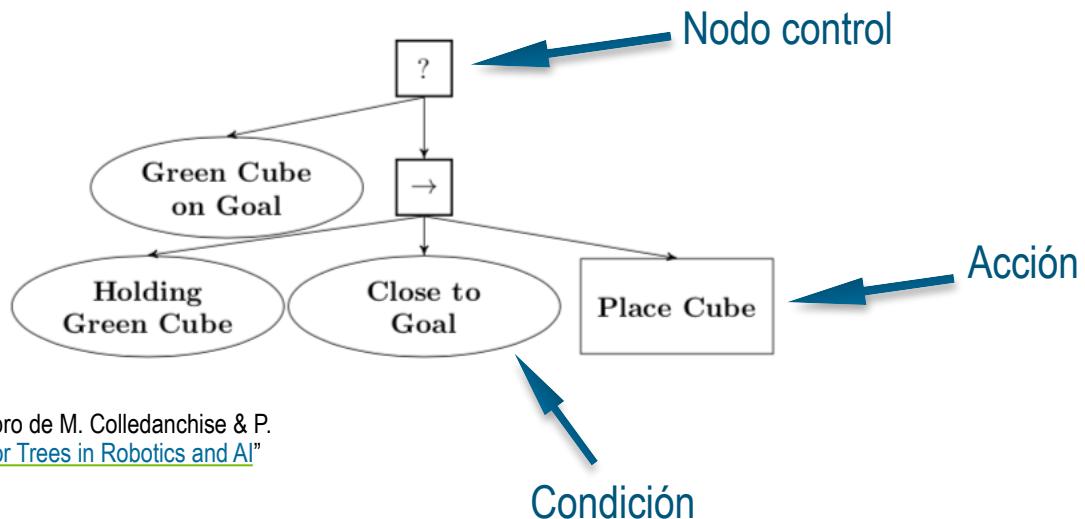
Fragmento del [behavior tree](#) de Halo 2



Behavior trees en Unity

# Behavior trees

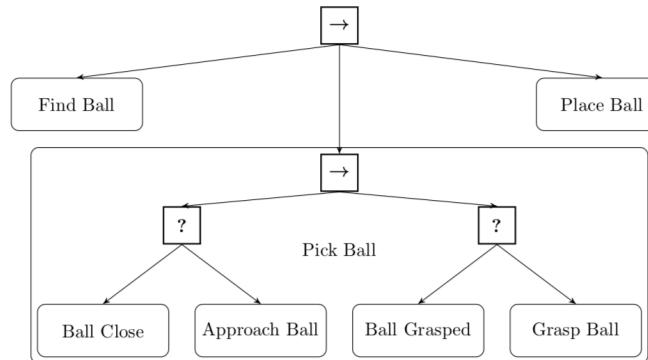
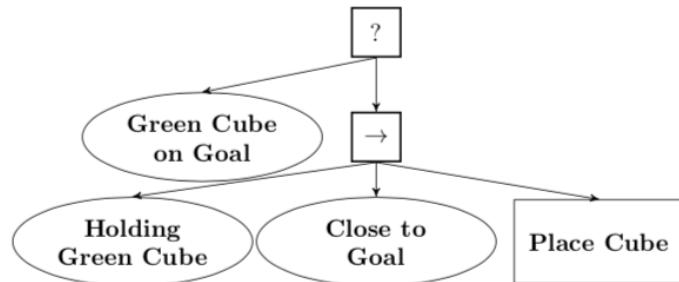
- Los nodos pueden ser condiciones, acciones o nodos de control
- Un nodo hoja puede tener 3 estados : {*success*, *failure*, *running*}



Ejemplos del libro de M. Colledanchise & P. Ögren "[Behavior Trees in Robotics and AI](#)"

# Nodos de control básicos

- **Selector (?)**: Representa alternativas.
  - Se queda en el primer nodo hijo que tenga éxito, (o esté *running*)
- **Secuencia(→)**: serie de pasos en la que todos son necesarios
  - Si un nodo hijo falla, termina en fallo (idem running).
  - Solo tiene éxito si todos lo tienen

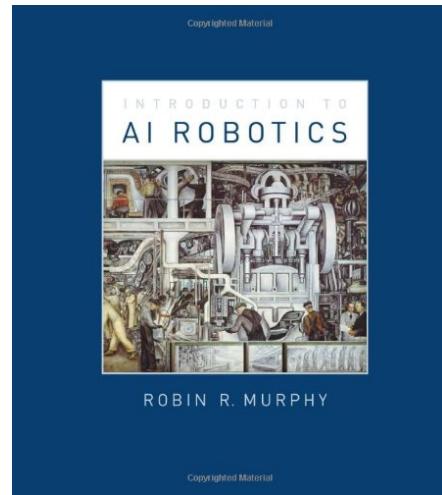


# Conclusiones

- La robótica reactiva y basada en comportamientos ha tenido bastante **éxito en dominios limitados**
- Constituye la **base de las arquitecturas híbridas** deliberativas/reactivas que veremos en sesiones sucesivas

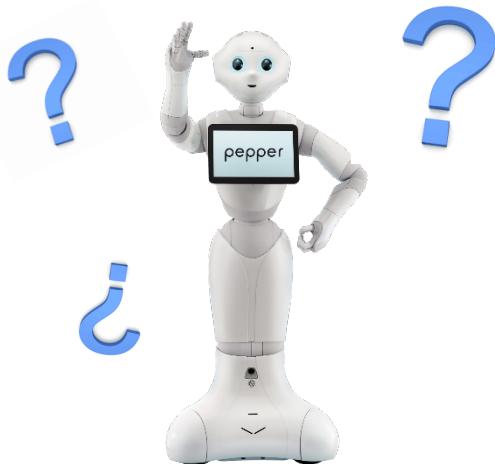
# Referencias

- **Introduction to AI Robotics,**  
Robin R. Murphy, MIT Press,  
2000



# Robots Móviles

Grado en Ingeniería Robótica



otto @ dccia.ua.es

