
ROBÓTICA

Grupo de Sistemas y Comunicaciones

jmplaza@gsync.es



Curso 2007-2008

Sistemas basados en comportamientos (SBC)

Introducción

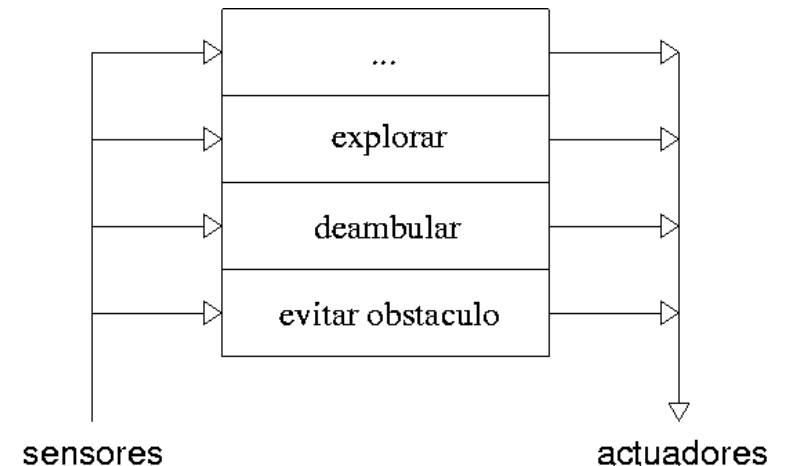
- Objetivo: que un robot se comporte inteligentemente.
- Programa monolítico secuencial: sensor-modelar-pensar-actuar.
 - **Descomposición funcional** típica de sistemas deliberativos.
 - Unos módulos llaman a funciones de otros.
 - Cuello de botella.
 - Escasez de buenos resultados prácticos, con robots reales.
- Sistemas reactivos puros no escalan en complejidad.
- SBC apuestan por un control distribuído.
 - **Descomposición en comportamientos.**
 - La actuación inteligente emerge de la interacción entre ellos.
 - Ejecución concurrente.

¿Qué es un comportamiento?

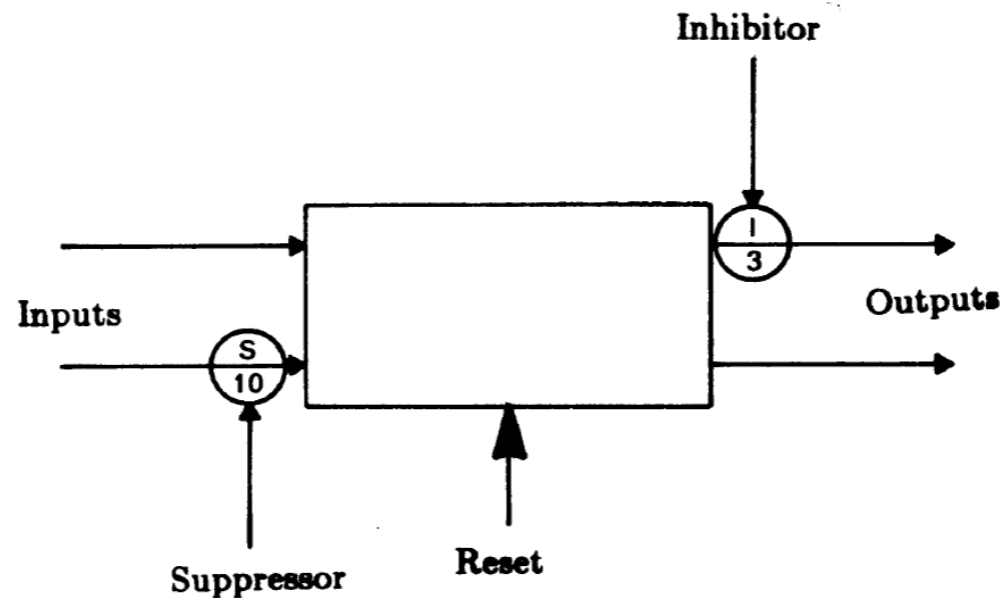
- Cada uno tiene un **objetivo** concreto (ir a casa, encontrar un socio, evitar a otros, etc.) y trata de conseguirlo.
- Suele ser un controlador realimentado.
- Pueden almacenar estado y utilizar representaciones del mundo.
- Pueden conectar directamente sensores con actuadores.
- Pueden recibir/enviar información de/a otros comportamientos.
- Los comportamientos se pueden “escribir” de diversas formas: funcional, diagramas SR, máquinas de estado finito, etc.
- Varios nombres: comportamiento, controlador, agente, esquema.

Arquitectura de subsunción de Brooks

- 1986, Rodney Brooks MIT.
- Descomposición en **niveles de competencia**.
- Cada nivel se ejecuta independiente y concurrentemente con el resto.
- Cada uno lee sensores y emiten actuación.
- Se materializa como automatas finitos de estado.
- Los inferiores se ocupan de la funcionalidad básica.



Subsunción

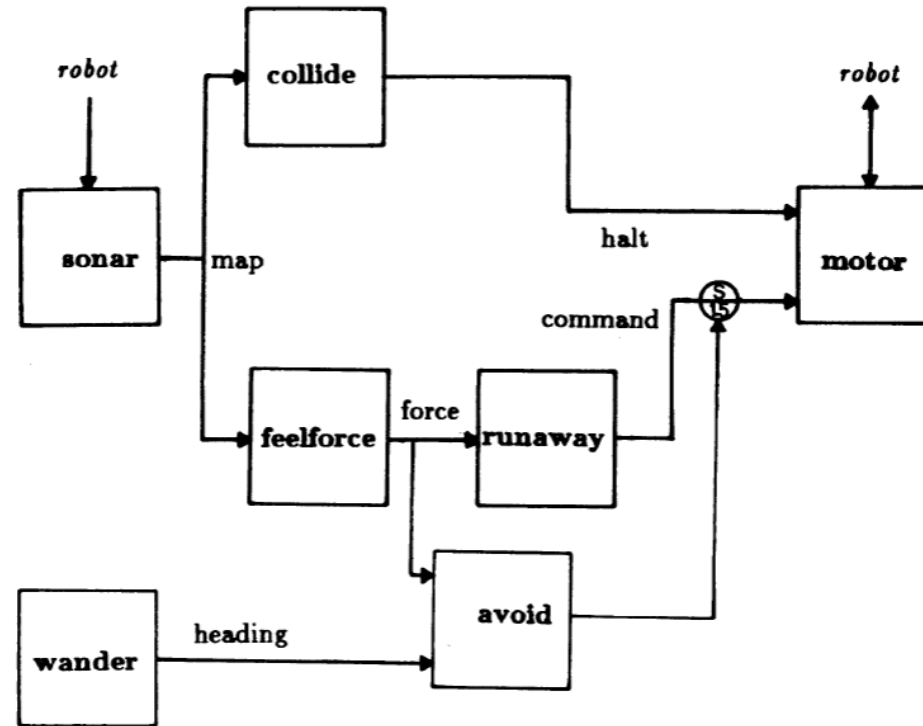


- Cada nivel se prueba a fondo en el entorno real.
- Desarrollo incremental: niveles superiores se construyen sobre los inferiores.
- Suprimiendo sus salidas y/o subsumiendo (suplantando) sus entrada.

Implementación con Autómatas

- Los niveles de competencia se materializan con máquinas de estado finito.
- A las máquinas se les añadió un pequeño estado: AFSM (*Augmented Finite State Machines*)
- Se implementaron en lisp.
- Cada AFSM puede estar en un estado en cada momento, recibir una o más entradas y producir una o más salidas.
- Las AFSM están conectadas para transmitirse mensajes.
- Las AFSM se organizan en redes, y estas en niveles.

Un ejemplo



- Nivel 0: sonar, collide, feelforce, runaway, motor.
- Nivel 1: wander, avoid.

Ejemplo: El robot Herbert

- Busca lata, la coge, la levanta, la pesa, si está llena la deja, si no la estruja y se la lleva.
- Usa visión y un láser para detectar las latas e infrarrojos y sonar para moverse.
- Jonathan Connell usó un procesador para implementar cada AFSM (1991).
- No mantiene estado sobre que ha hecho y cual es lo siguiente.
- Sólo percibe.
- No hay relación directa entre las AFSM que se encargan de cada función.

Ejemplo: El robot Toto (Mataric)

- Construye mapas topológicos.
- Comportamiento detecta marca.
- Comportamientos de navegacion:

```
(defbehavior align
  (cond
    ((and (< (sonar 7 or 8) edging_distance)
          (> (sonar 5 or 6) edging_distance))
     (turn right))

    ((and (< (sonar 9 or 10) edging_distance)
          (> (sonar 11 or 0) edging_distance))
     (turn left))))
```

Representación en Brooks

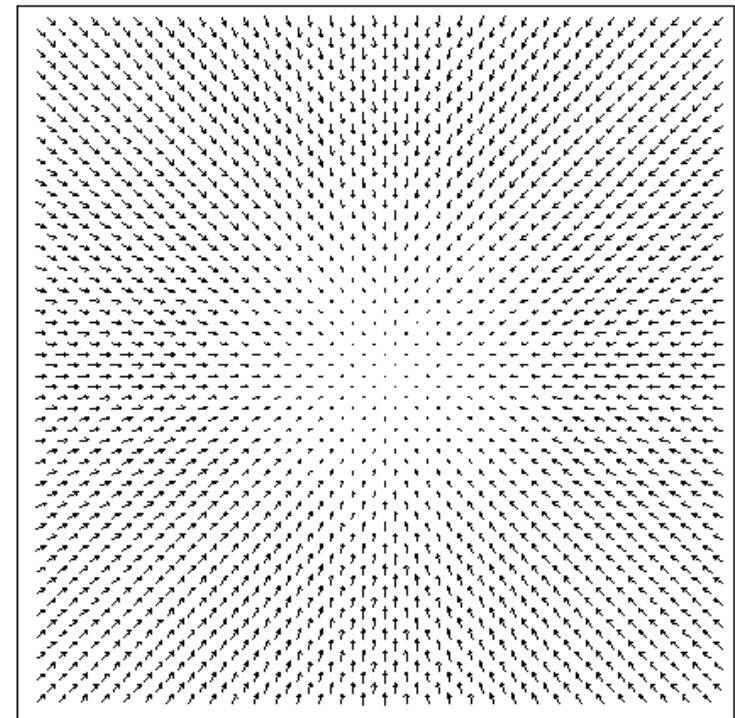
- No hay un modelo centralizado interno donde se fusiona toda la información sensorial.
- Cada comportamiento tiene su percepción en los sensores que le interesan.
- El control se emite desde los sensores directamente.
- No se necesitan modelos internos, ni representación simbólica alguna: **el mundo es su mejor modelo.**

Comportamiento emergente

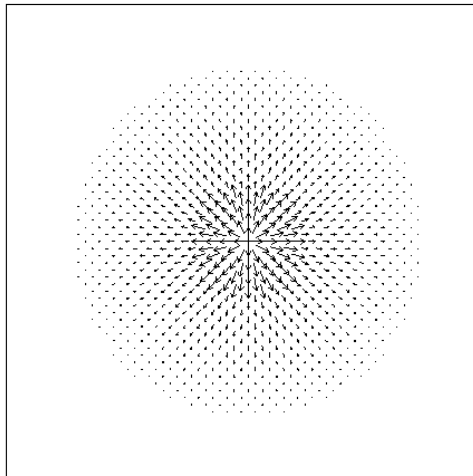
- El comportamiento surge de la interacción con el entorno.
- Por ejemplo, comportamiento seguir una pared:
 - Si estás lejos acercate.
 - Si estás muy cerca alejate.
 - En otro caso sigue.
- El robot no tiene el concepto de “pared” .
- El robot no tiene el concepto de “seguir” .

Esquemas

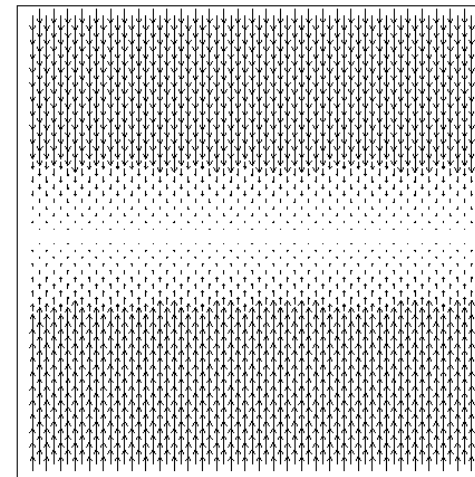
- Michael Arbib, Ronald Arkin.
- Fundamento biológico.
- Descomposición en **esquemas**.
- Esquemas perceptivos y esquemas motores.
- Un esquema es un patrón de acción.
- Su salida es un campo de potencial.



Ejemplo: piloto en presencia de obstáculos

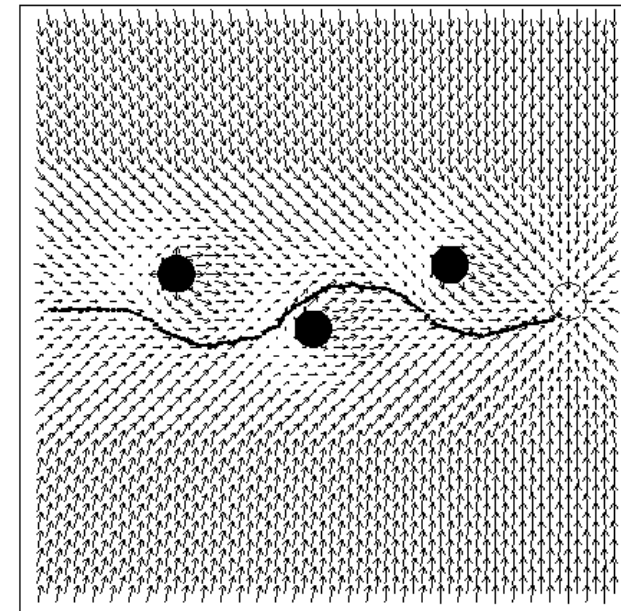


- Evitar obstáculo.
- Se instancia un esquema por obstáculo.



Mantenerse en el camino.

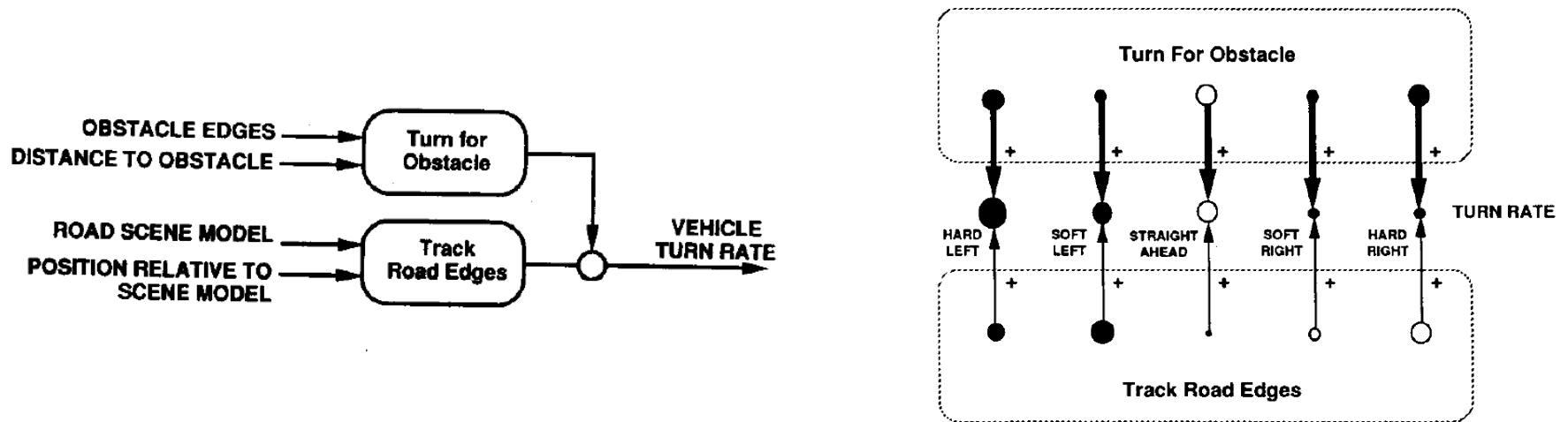
- El valor final es la **superposición** de los campos de potencial de cada uno de los esquemas (suma ponderada).
- Cada esquema motor lleva asociado algún esquema perceptivo que produce la información relevante para su objetivo.



Coordinación entre comportamientos

- ¿Que actuación es la más conveniente en este instante?
- La distribución del control obliga a resolver contradicciones entre comportamientos.
- Arbitraje: un único comportamiento elije.
 - prioridades.
 - basado en estados.
 - red de activación.
- Fusión de comandos:
 - suma ponderada.
 - fusión borrosa.
 - votación.

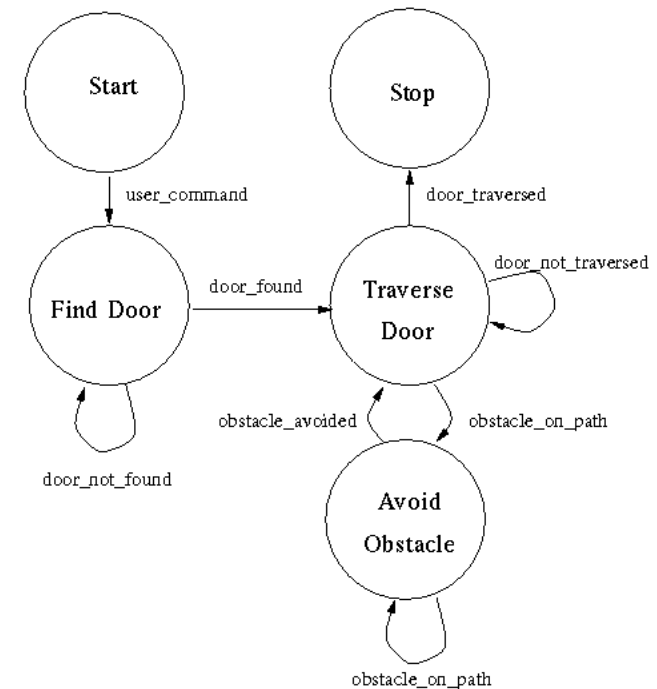
Fusión con votación de Payton



- Cada comportamiento expresa su grado de preferencia sobre todas y cada una de las opciones de actuación posibles.
- Aquella opción de consenso que es la que finalmente se envía a los actuadores.

Coordinación por Automata Finito (Arkin)

- Se definen varios estados relevantes.
- También los eventos que hacen saltar de uno a otro.
- En cada estado se activan ciertos esquemas.



Resumen

- Control distribuído en comportamientos.
- Exhiben alta reactividad.
- Percepción distribuída orientada a tarea, no simbólica.
- No mantienen un estado global del robot.
- Se han conseguido comportamientos complejos con robots reales.
- Difícil asignar y cambiar objetivos.
- La coordinación de comportamientos es complicada.