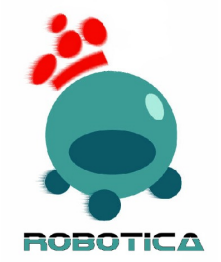
Arquitectura JDE para generar comportamientos en robots con visión

José María Cañas Plaza

http://gsyc.escet.urjc.es/jmplaza



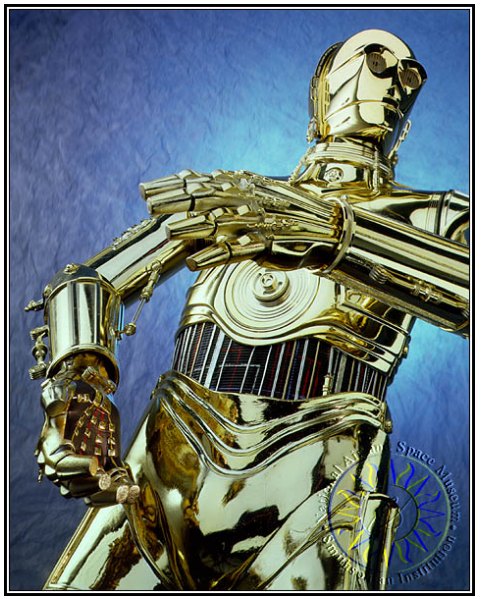
INAOE, Puebla (México), julio 2005

Contenidos

- Introducción
- Arquitectura JDE
- Implementación software
- Experimentos
- Conclusiones

Introducción

Robótica ficción vs Robótica real



Robótica real

¿Qué es un robot?

Sistema informático con:

- Sensores
- Actuadores
- Computador

Hay que programarlo para que consiga sus objetivos y sea sensible a la situación

¿Por qué no hay robots haciendo las tareas de casa?

- Generar comportamiento autónomo es difícil
- Integrar un conjunto amplio de conductas es muy complejo
- Falta robustez y flexibilidad
- ¿Problema tecnológico o teórico?
- Los animales se comportan inteligentemente en su entorno

Arquitectura de un robot

La organización de las capacidades sensoriales, de procesamiento y de acción para conseguir un repertorio de comportamientos inteligentes interactuando con cierto entorno

- Repertorio de comportamientos
- Información desbordante, incierta
- Selección de acción, atención
- La arquitectura determina el comportamiento observable

Diferentes paradigmas arquitectónicos

- Escuela deliberativa
- Escuela reactiva
- Sistemas basados en comportamientos
- Sistemas híbridos
- Etología: modelos comportamiento en animales

La etología puede aportar a la robótica modelos para generar comportamientos más complejos e integrarlos en un único sistema

Arquitectura JDE

Hipótesis de partida

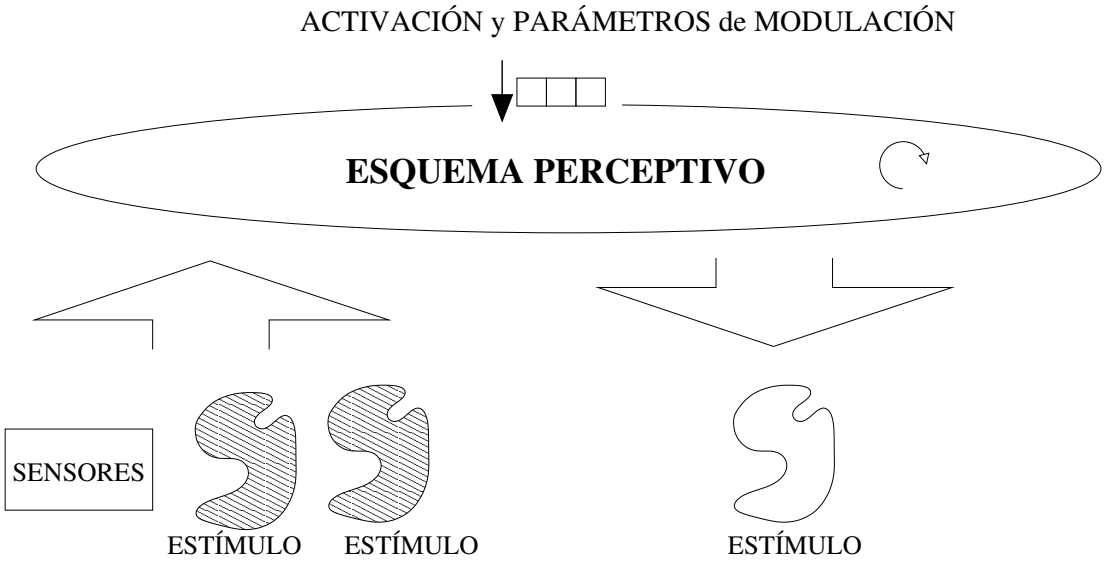
- Comportamiento = percepción y control
- Fragmentación en unidades (esquemas) asíncronas concurrentes
  - de percepción elaboran estímulos
  - de actuación toman decisiones
- La colección de esquemas se organiza en jerarquía
- Jerarquía Dinámica de Esquemas

Esquemas

Un esquema es un flujo de ejecución independiente, con un objetivo

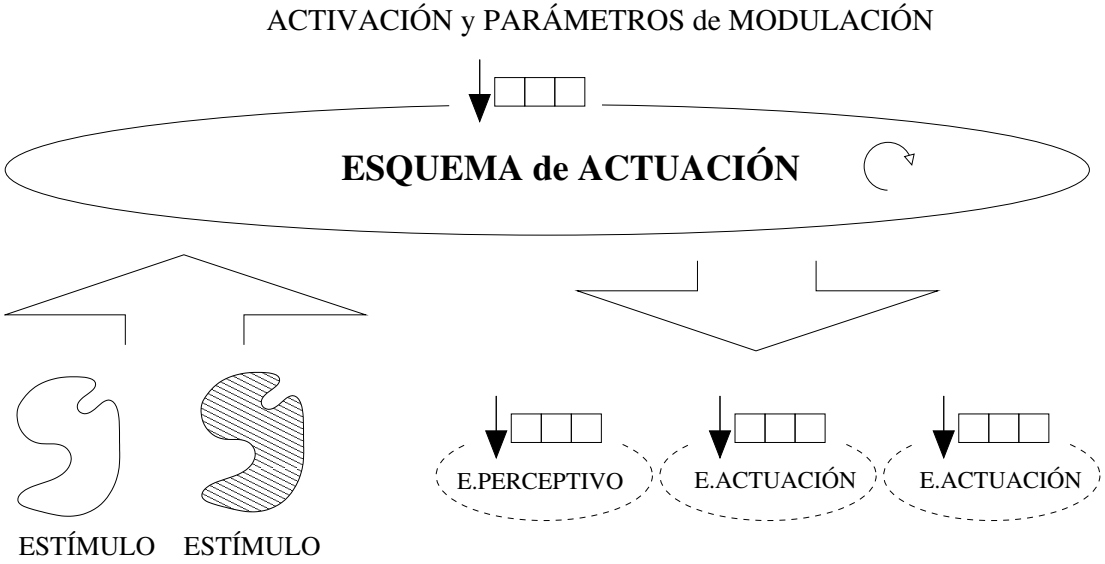
- Iterativo
- Se puede activar y desactivar a voluntad
- Modulable a través de parámetros
- Esquemas perceptivos y de actuación
- Su funcionalidad se usa despertándolo y modulándolo

Esquemas perceptivos



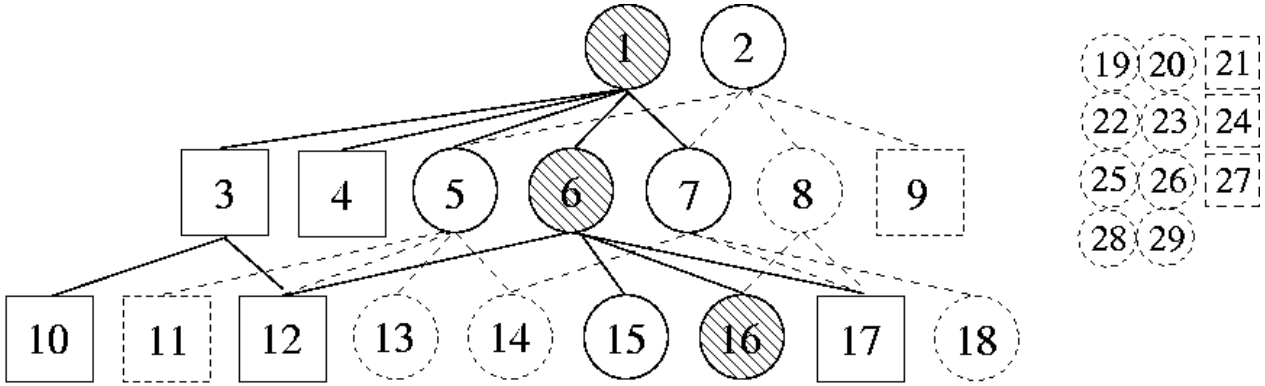
- Producen estímulos y los mantienen actualizados (anclados)
- Simples lecturas sensoriales, transformaciones más elaboradas o depender de otros estímulos menos abstractos
- Estados: DORMIDO o ACTIVO

Esquemas de actuación



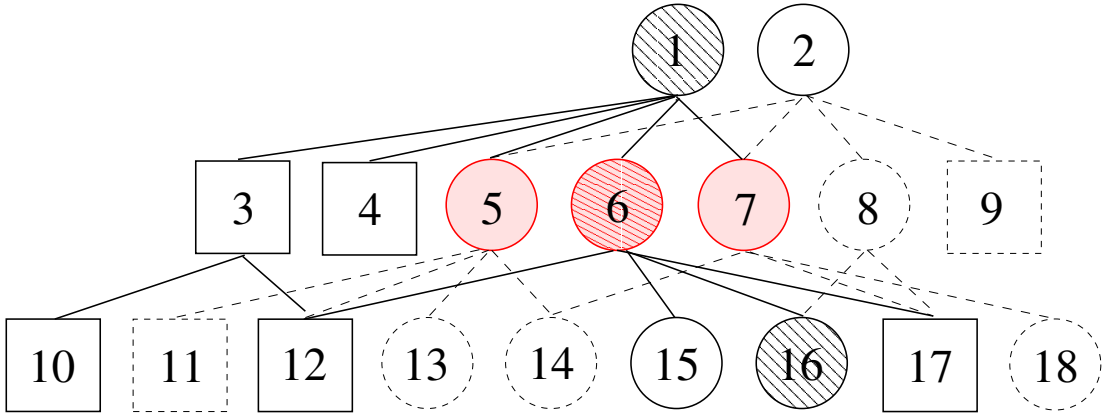
- Toman decisiones de actuación para conseguir o mantener un objetivo
- Su salida pueden ser comandos a los actuadores o la activación y modulación de otros esquemas hijos
- Estados: DORMIDO, ALERTA, PREPARADO o ACTIVO

Jerarquía

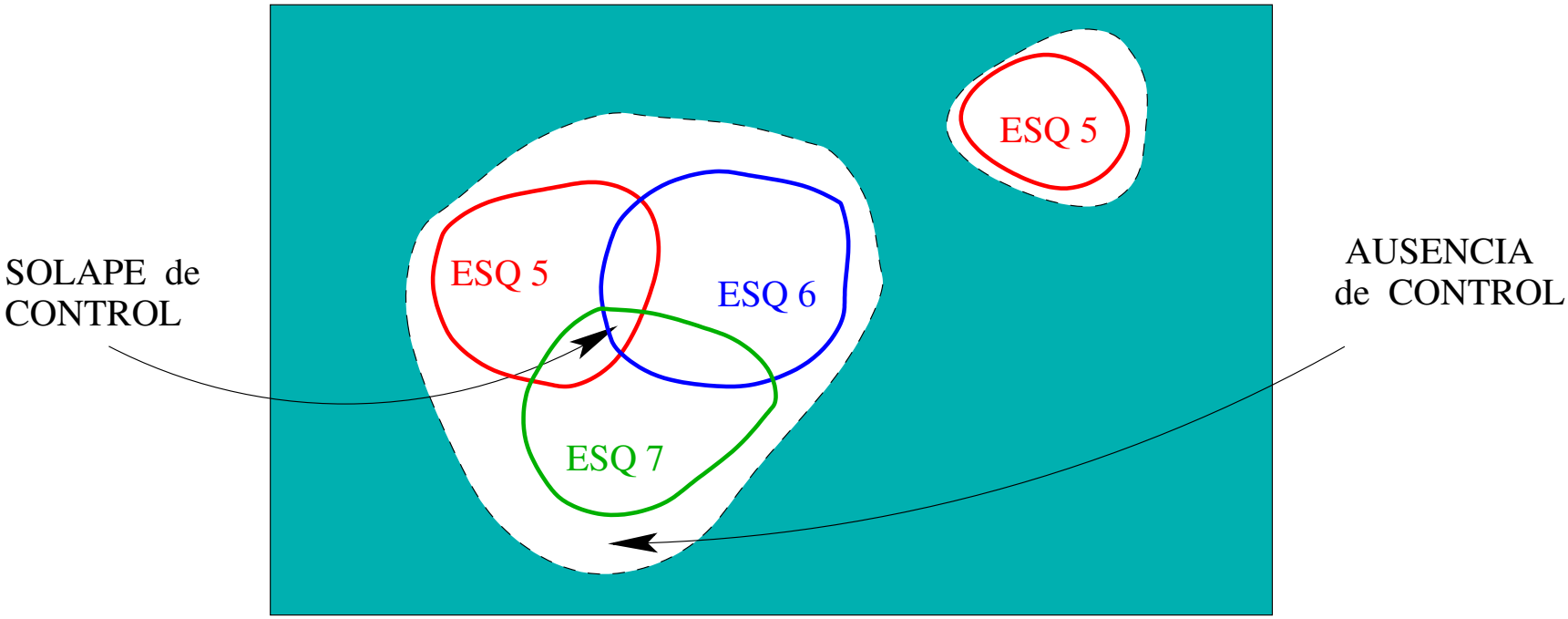


- Los esquemas despiertos se ejecutan asíncrona y concurrentemente
- Los esquemas se estructuran en jerarquía, donde hay padres e hijos
- Se construye y reconfigura dinámicamente
- Es específica de cada comportamiento
- El número de niveles depende de la complejidad de la tarea

Selección de acción: ¿Qué hago ahora?



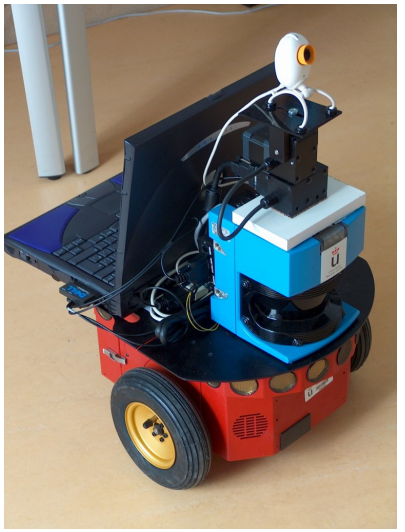
- Distribución: una competición por nivel
- El padre determina el paso de DORMIDO a ALERTA
- La situación determina el paso de ALERTA a PREPARADO
- De PREPARADO a ACTIVO se resuelve con un arbitraje



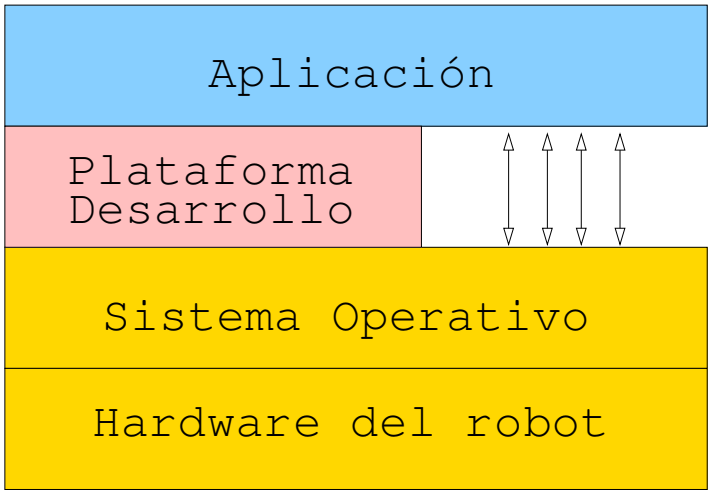
- Precondiciones configuran regiones de activación
- Jerarquía como predisposición: ALERTA $\neq$ ACTIVO
- Arbitraje explícito en colisiones y vacíos de control

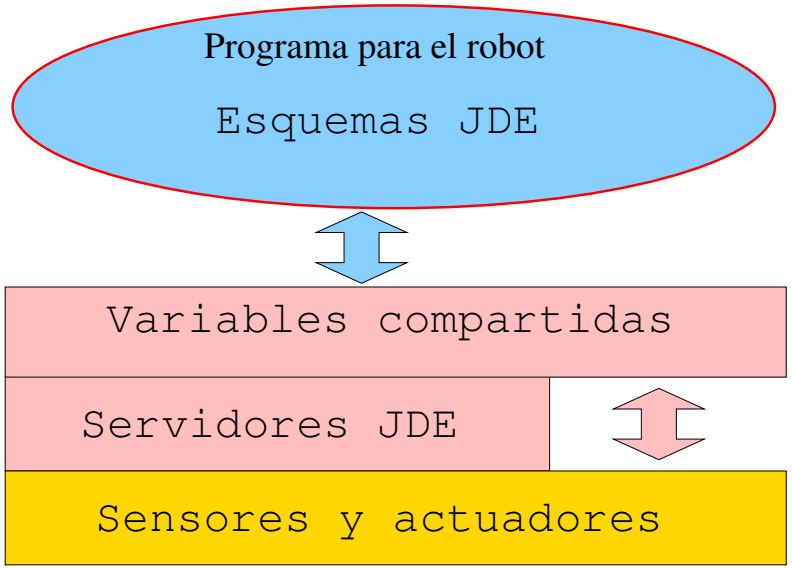
Implementación software de JDE: JDE.c

- La aplicación robótica es una colección de esquemas
  - Esquemas como hebras
  - Iteraciones periódicas de control
  - Leer y escribir variables
- JDE.c ofrece acceso al hardware a través de variables
  - Variables sensoriales
  - Variables de control
- Servidores de red para acceso remoto
- Robots reales (pioneer) y simuladores
- Lenguaje C



Programación con JDE.c





JDE.c básico

- Variables sensoriales:

  laser

  us

  x, y, theta

  colorA

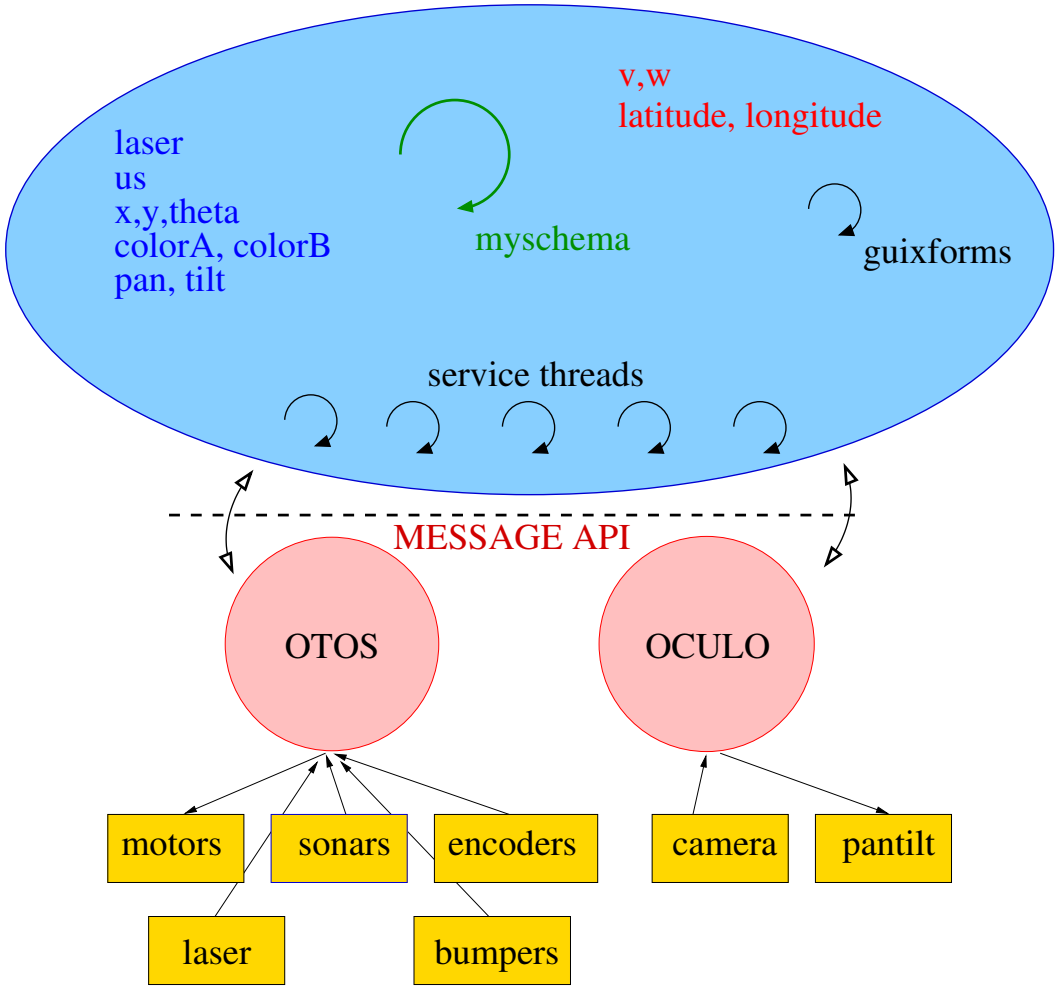
  colorB

  pan, tilt

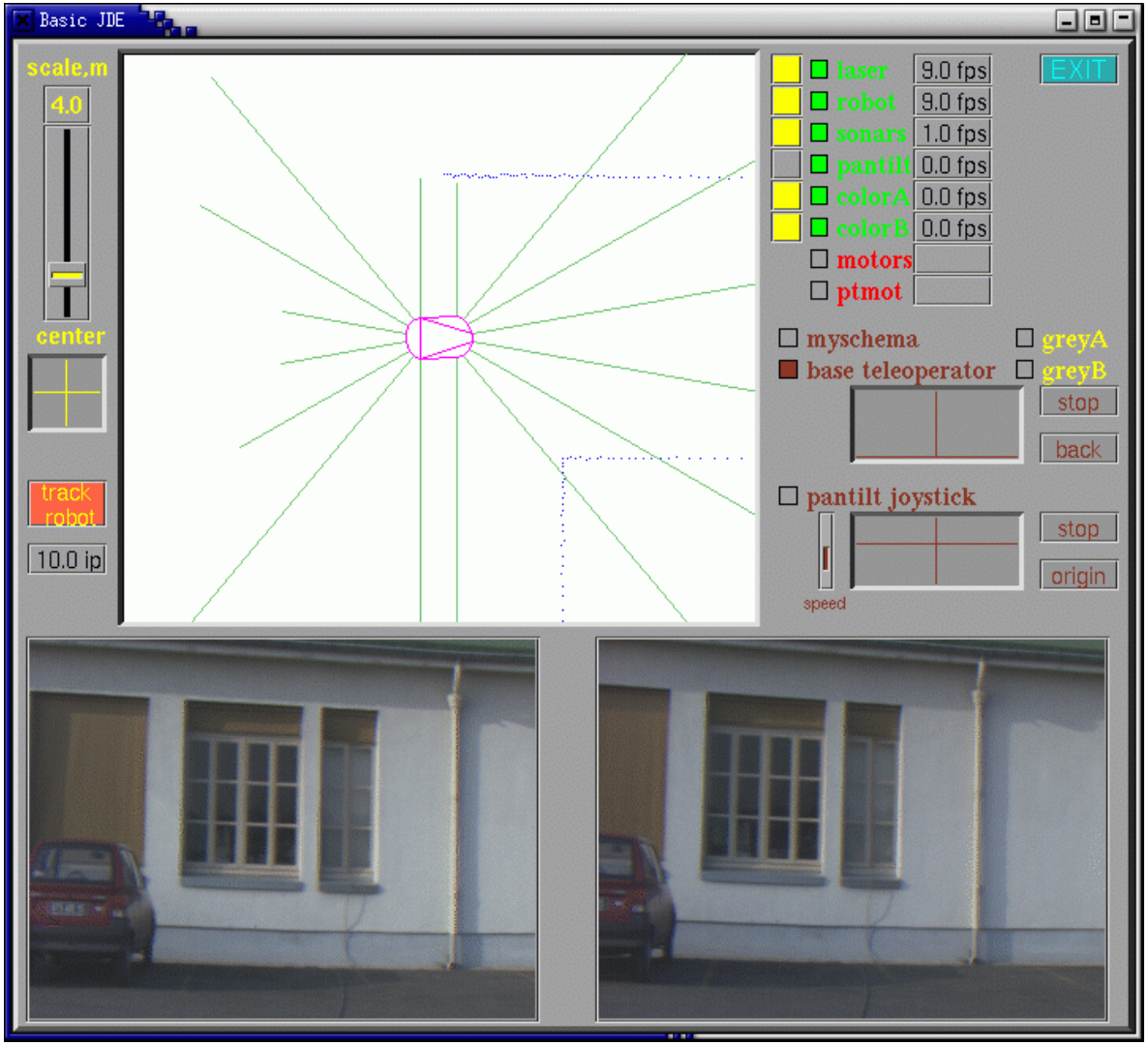
- Variables de actuación:

  v,w

  latitude, longitude



Guixforms


Basic JDE
scale,m
4.0
center
track robot
10.0 ip
laser 9.0 fps
robot 9.0 fps
sonars 1.0 fps
pantilt 0.0 fps
colorA 0.0 fps
colorB 0.0 fps
motors
ptmot
EXIT
myschema
base teleoperator
greyA
greyB
stop
back
pantilt joystick
speed
stop
origin

jde.conf

- Para cada dispositivo (sensor/actuador) especifica su fuente
- Fuente: local/remota, real/simulada

```
sonars    otos localhost 3000
laser    otos localhost 3000
encoders    otos localhost 3000
motors  otos localhost 3000
imageA  file casaleft.pnm
imageB  file casaright.pnm
#imageA  oculo localhost 3001 1
#imageA  video4linux /dev/video1
#imageB  firewire 0
```

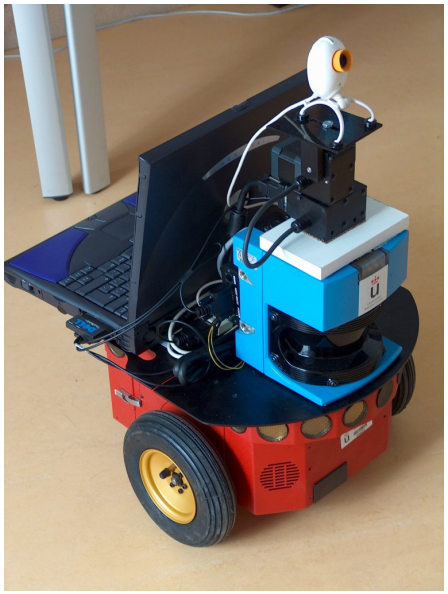
Esqueleto de un esquema

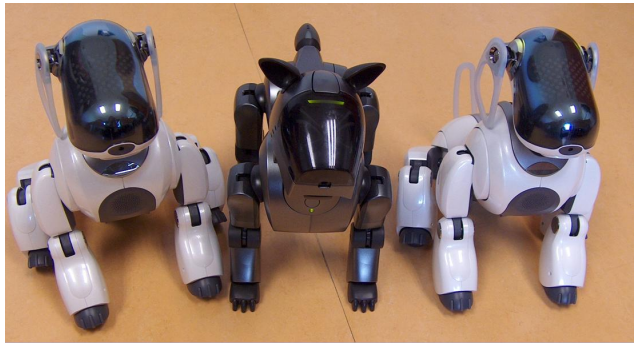
```
void *square_thread(void *not_used)
{
  struct timeval a,b;
  long diff, next;

  for(;;)
    {
      pthread_mutex_lock(&mymutex[SCH_SQUARE]);

      if (state[SCH_SQUARE]==slept)
        {
          printf("square: off\n");
          v=0; w=0;
          pthread_cond_wait(&condition[SCH_SQUARE],&mymutex[SCH_SQUARE]);
          printf("square: on\n");
          square_state=init;
        }
      else
```

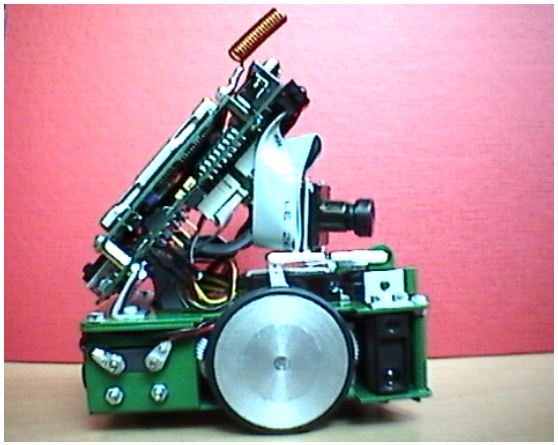
```
        {
          gettimeofday(&a,NULL);
          square_iteration();
          gettimeofday(&b,NULL);
          diff = (b.tv_sec-a.tv_sec)*1000000+b.tv_usec-a.tv_usec;

          next = square_cycle*1000-diff-10000;
          /* discounts 10ms taken by calling usleep itself */
          if (next>0) usleep(square_cycle*1000-diff);
          else {printf("time interval violated: square\n");
        usleep(square_cycle*1000);}
          }
      pthread_mutex_unlock(&mymutex[SCH_SQUARE]);
 }
}
```

Experimentos

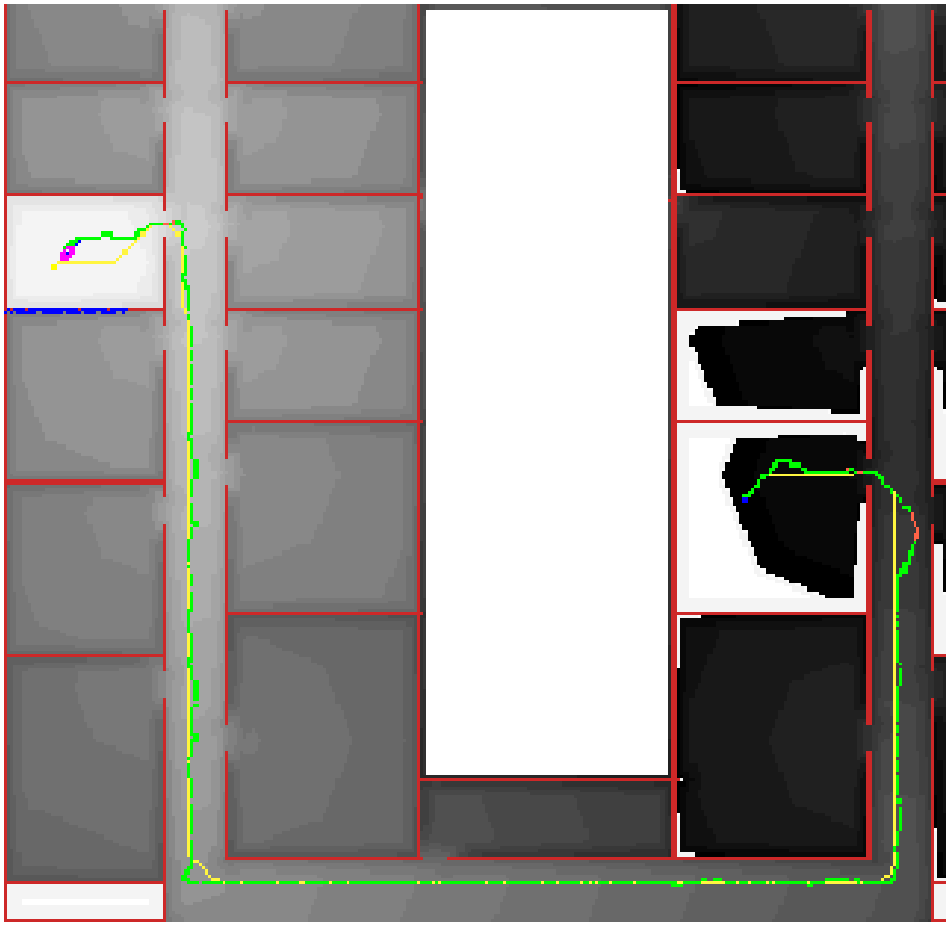
Robots reales y simuladores



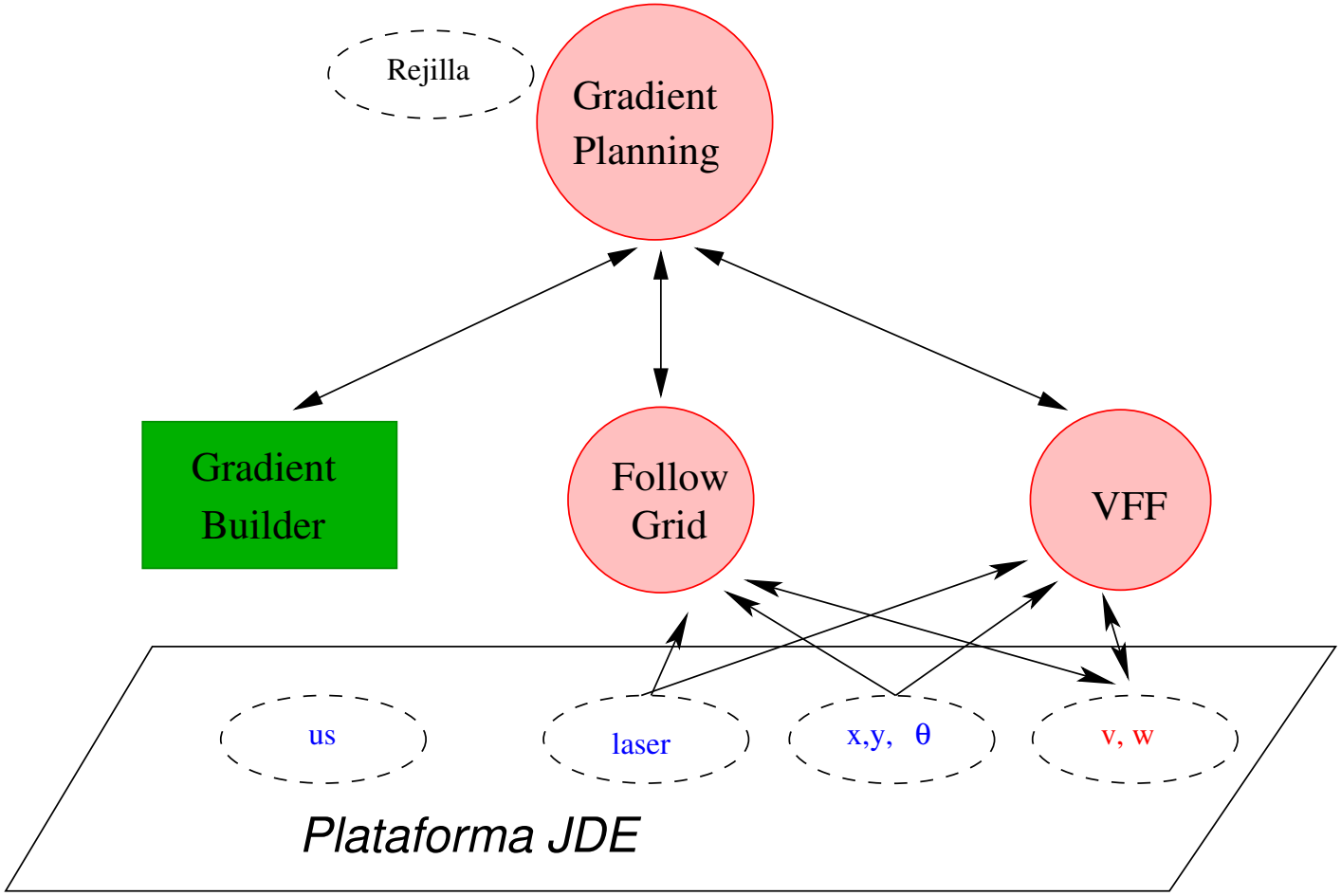




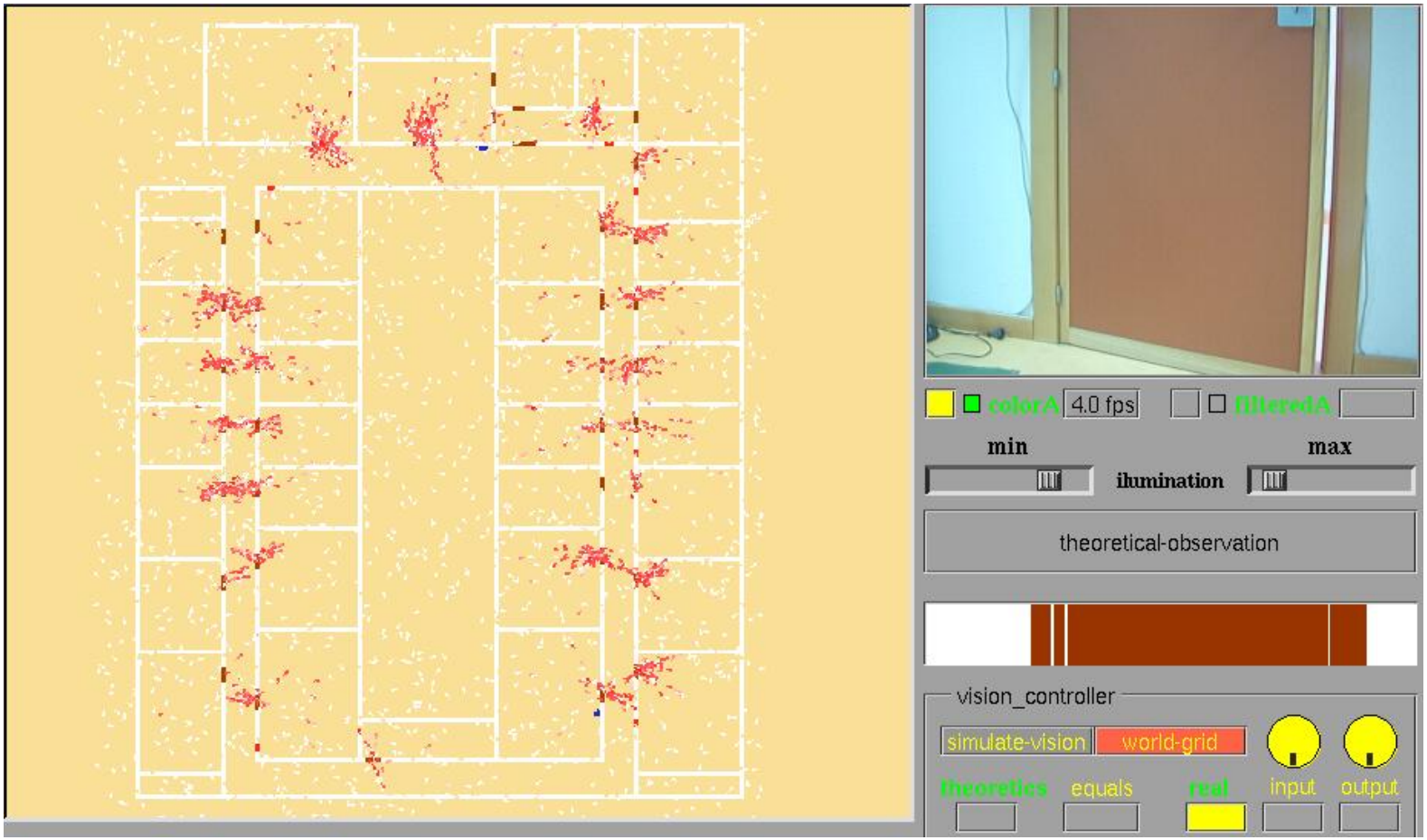
Navegación global GPP y VFF



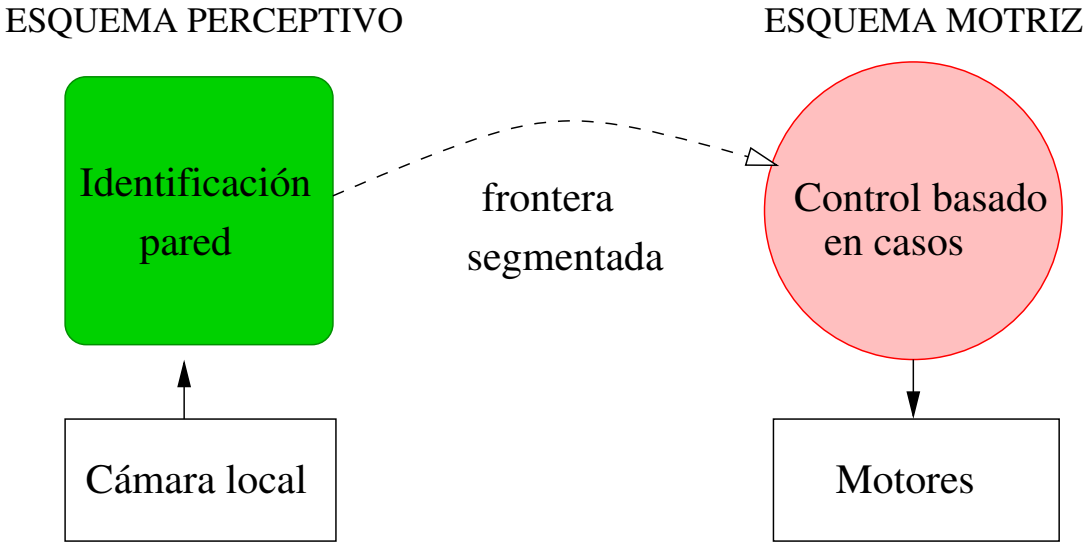
¿Cómo?

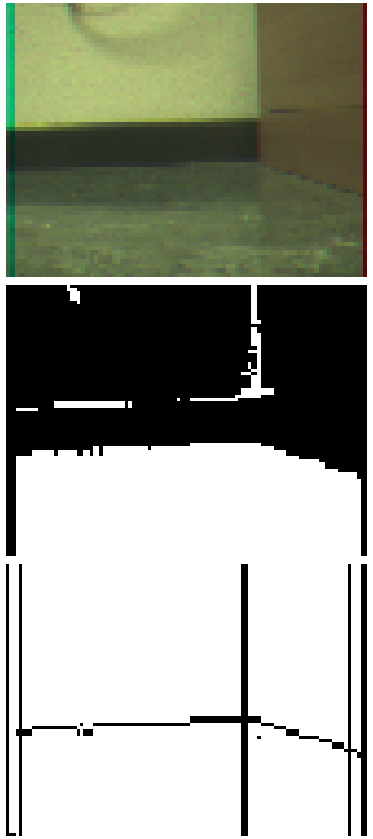

Rejilla
Gradient Planning
Gradient Builder
Follow Grid
VFF
us
laser
x,y, θ
v, w
Plataforma JDE

Esquema perceptivo localización visual

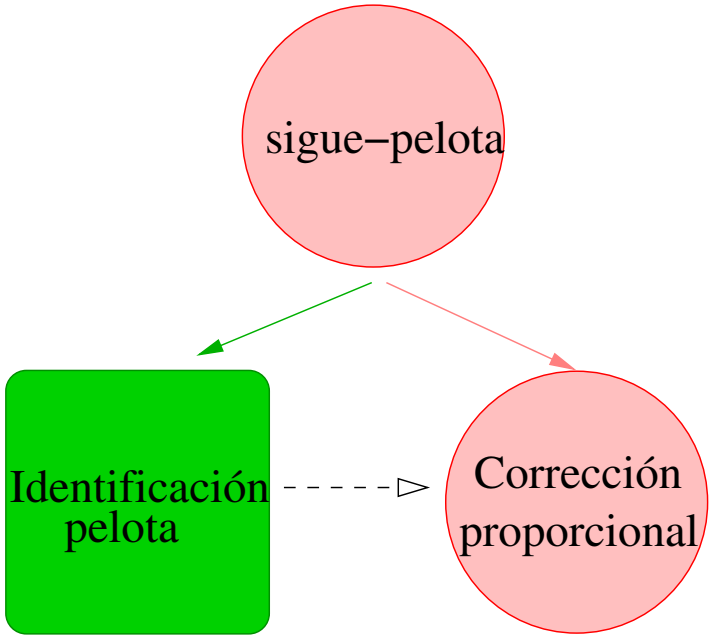


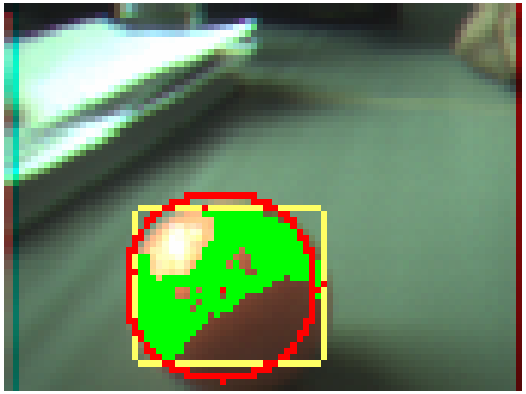
Taxias: comportamiento sigue-pared con visión

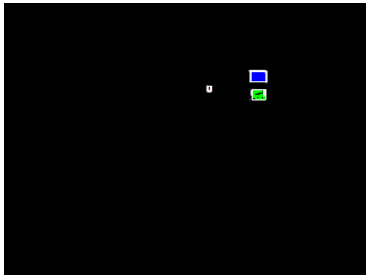




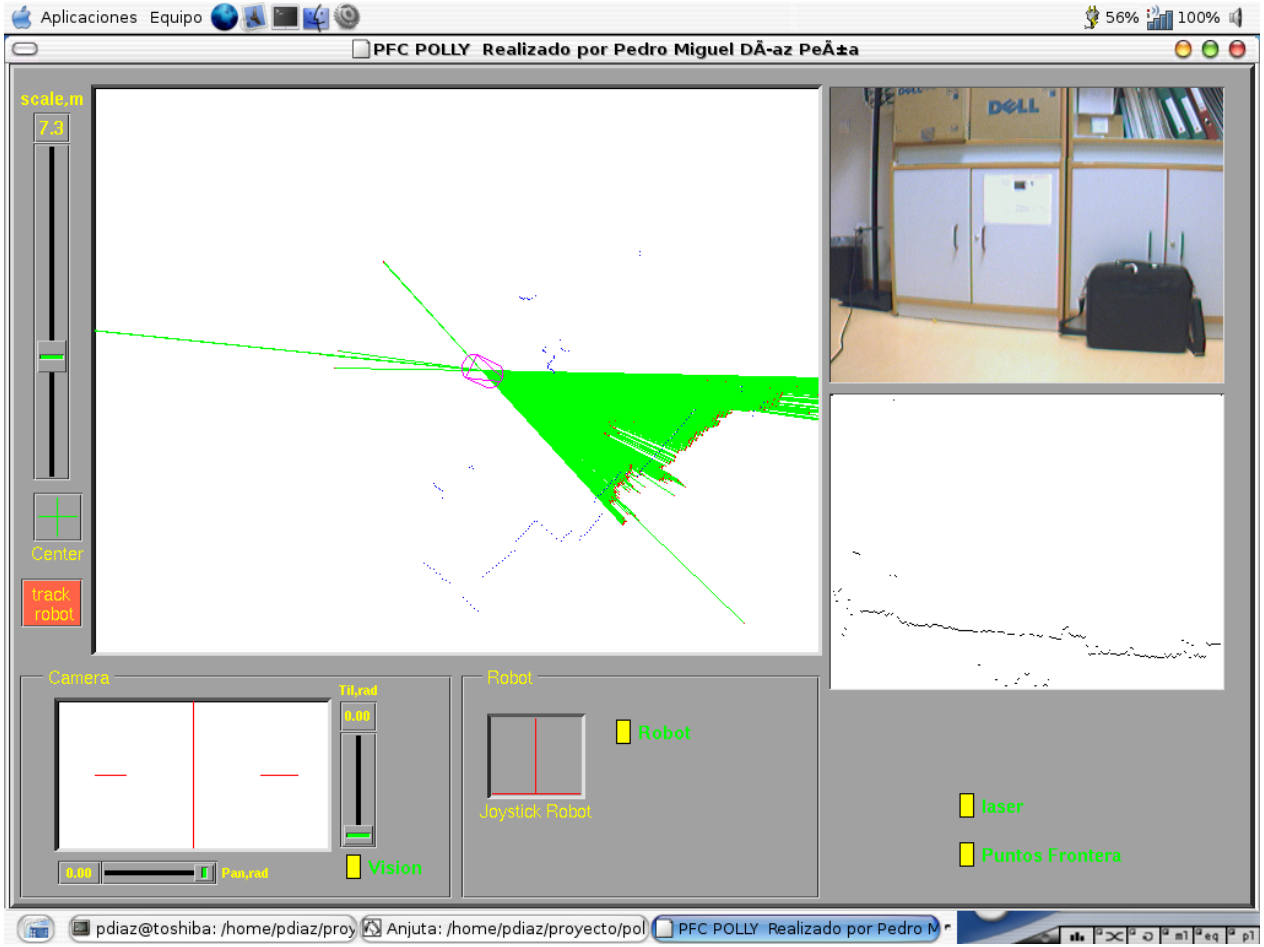
Taxias: comportamiento sigue-pelota con visión



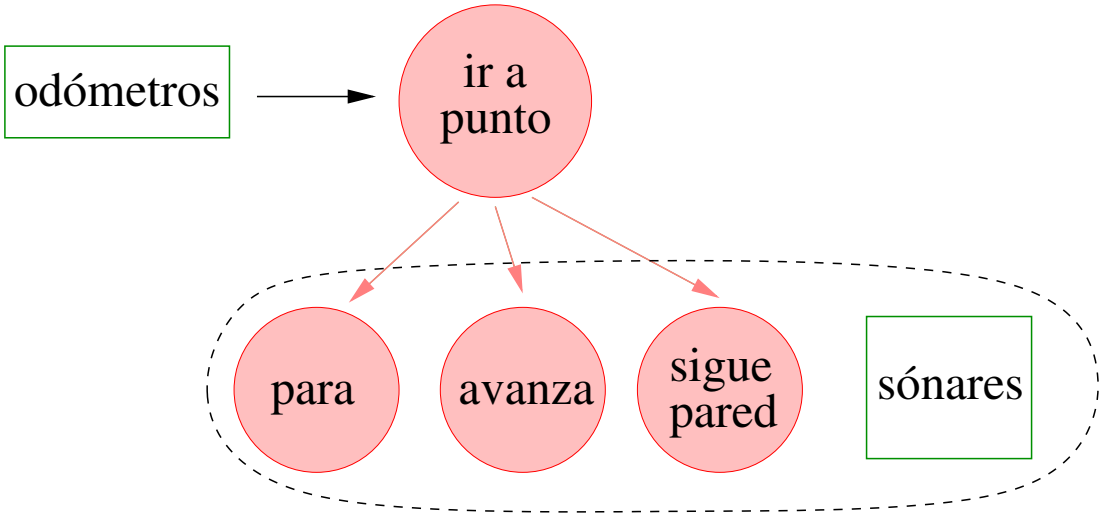


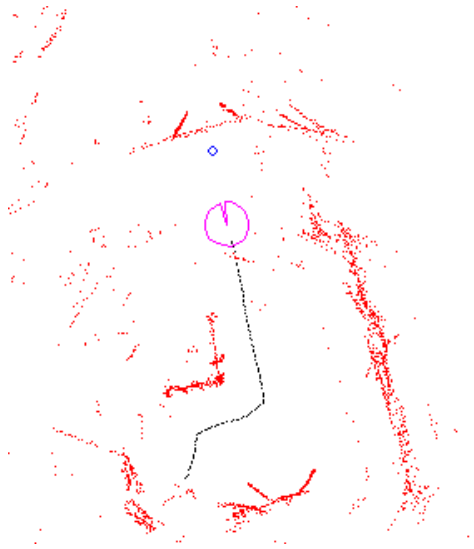


Navegación local con visión monocular

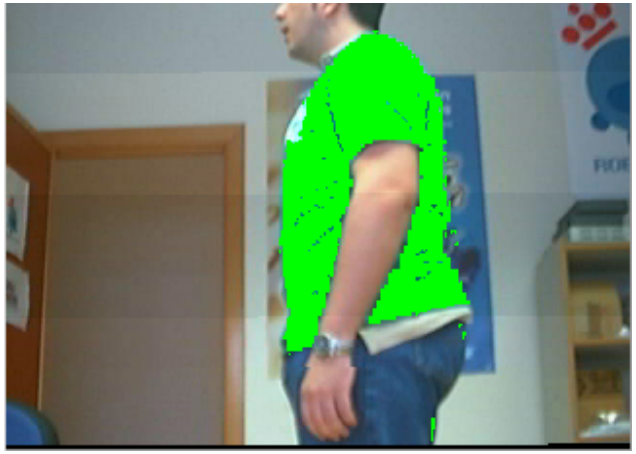


Jerarquías simples: comportamiento ir-a-punto

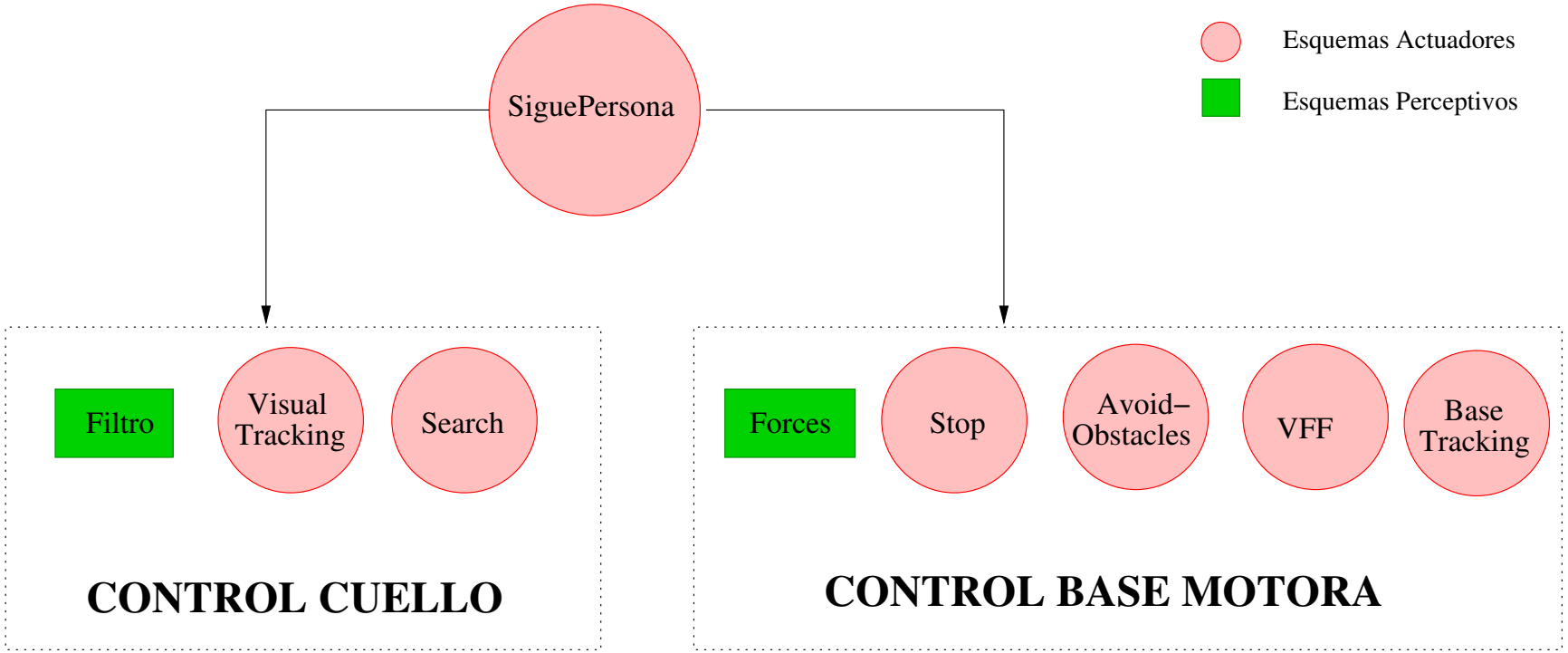




Comportamiento sigue-persona con visión

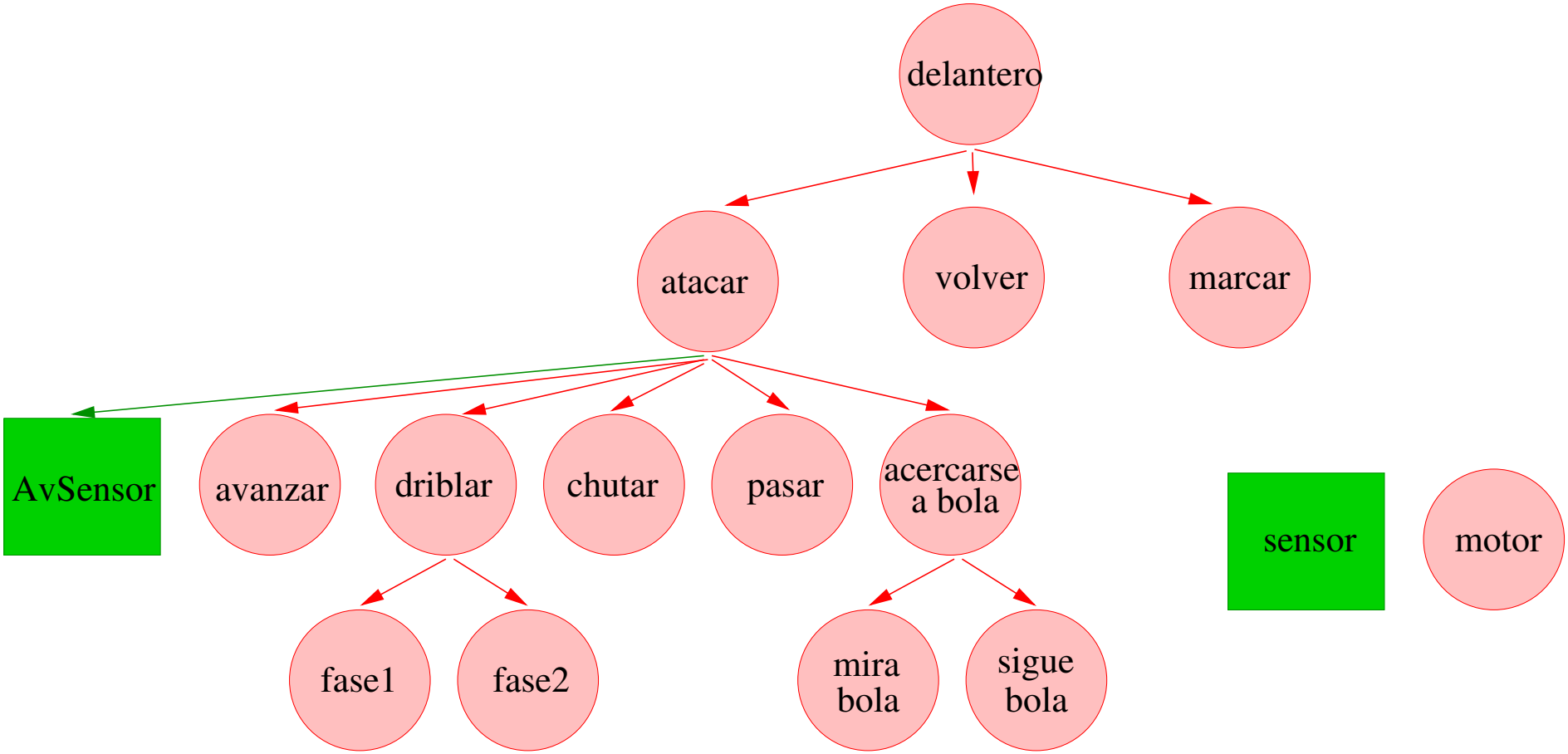


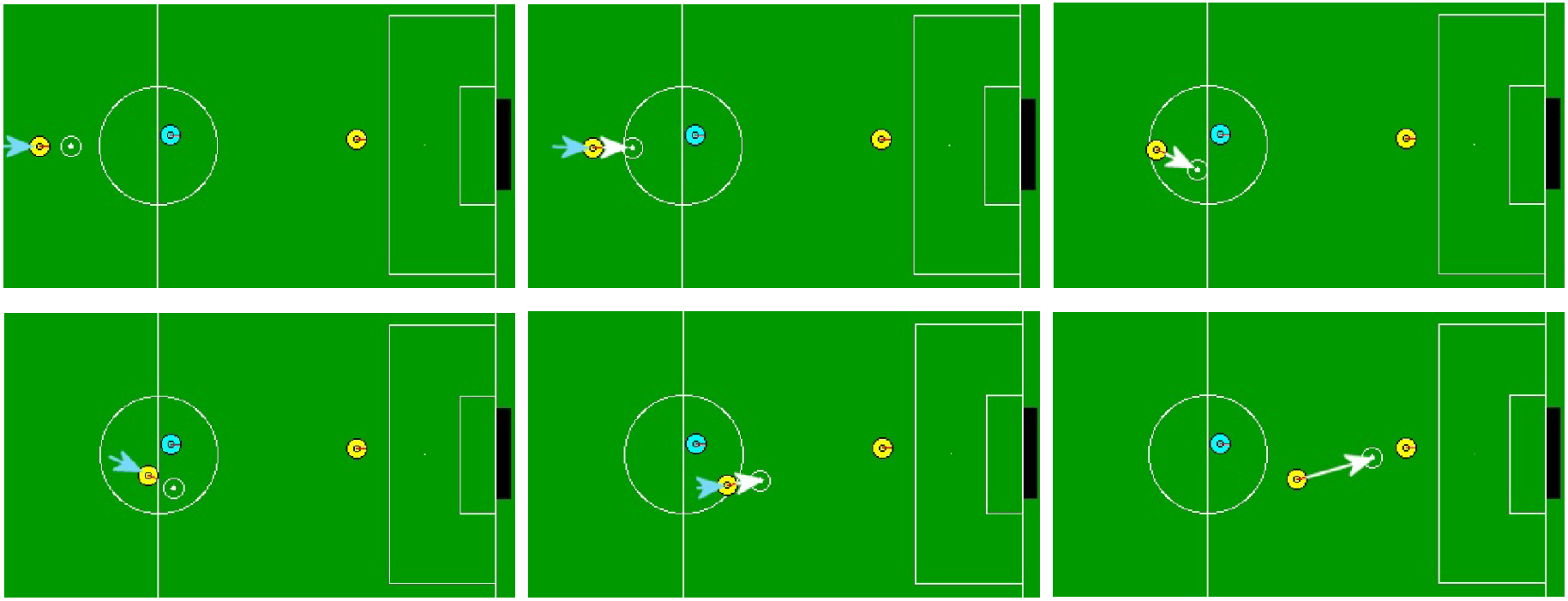
¿Cómo?


SiguePersona
Esquemas Actuadores
Esquemas Perceptivos
Filtro
Visual Tracking
Search
CONTROL CUELLO
Forces
Stop
Avoid–Obstacles
VFF
Base Tracking
CONTROL BASE MOTORA

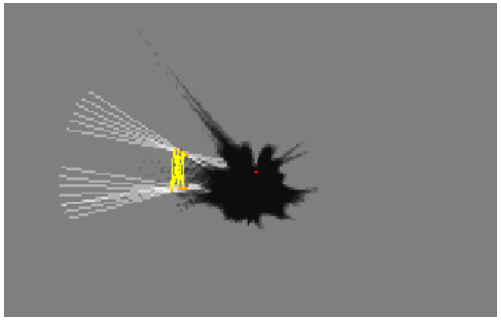
También en el perrito

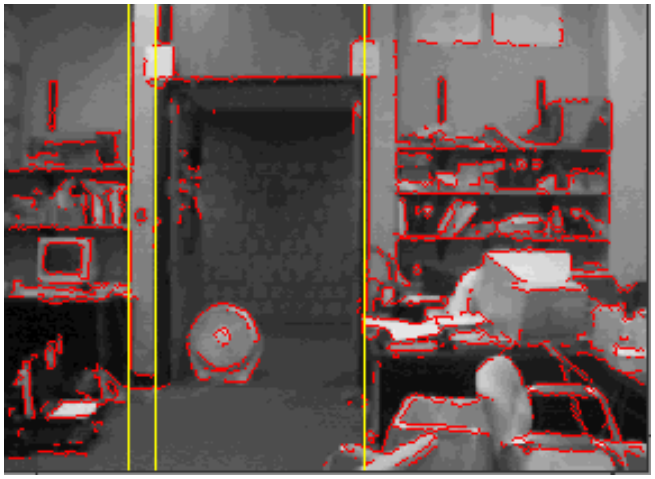
Jerarquías complejas: jugador de RoboCup

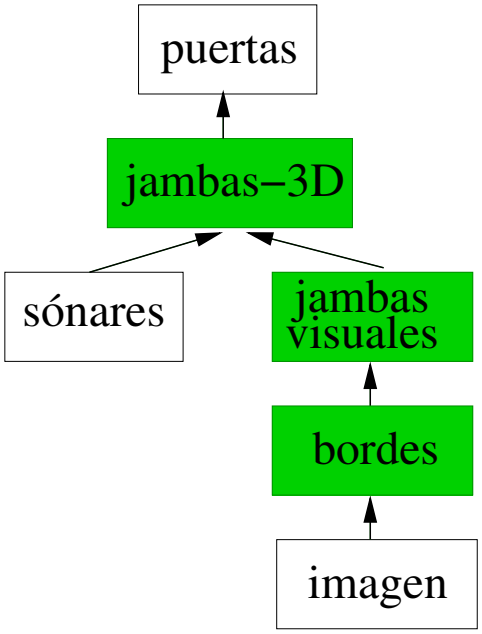




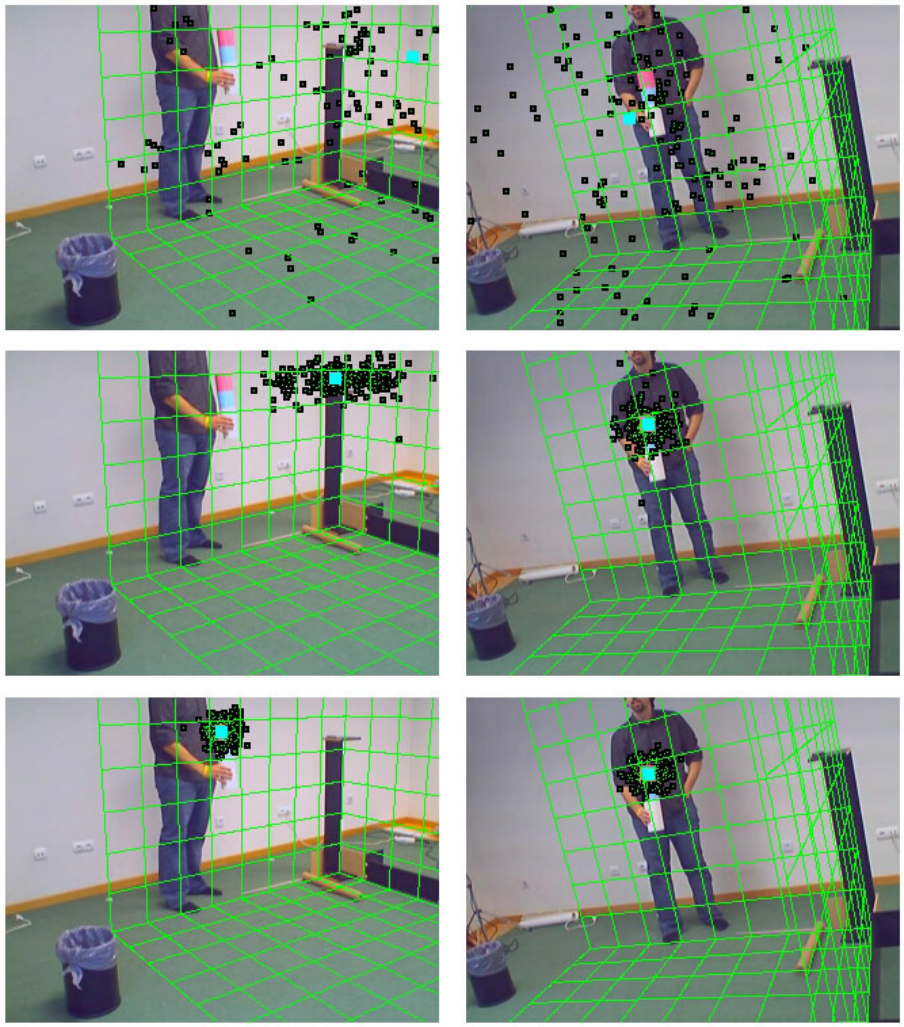
Visión avanzada: estímulos estructurados



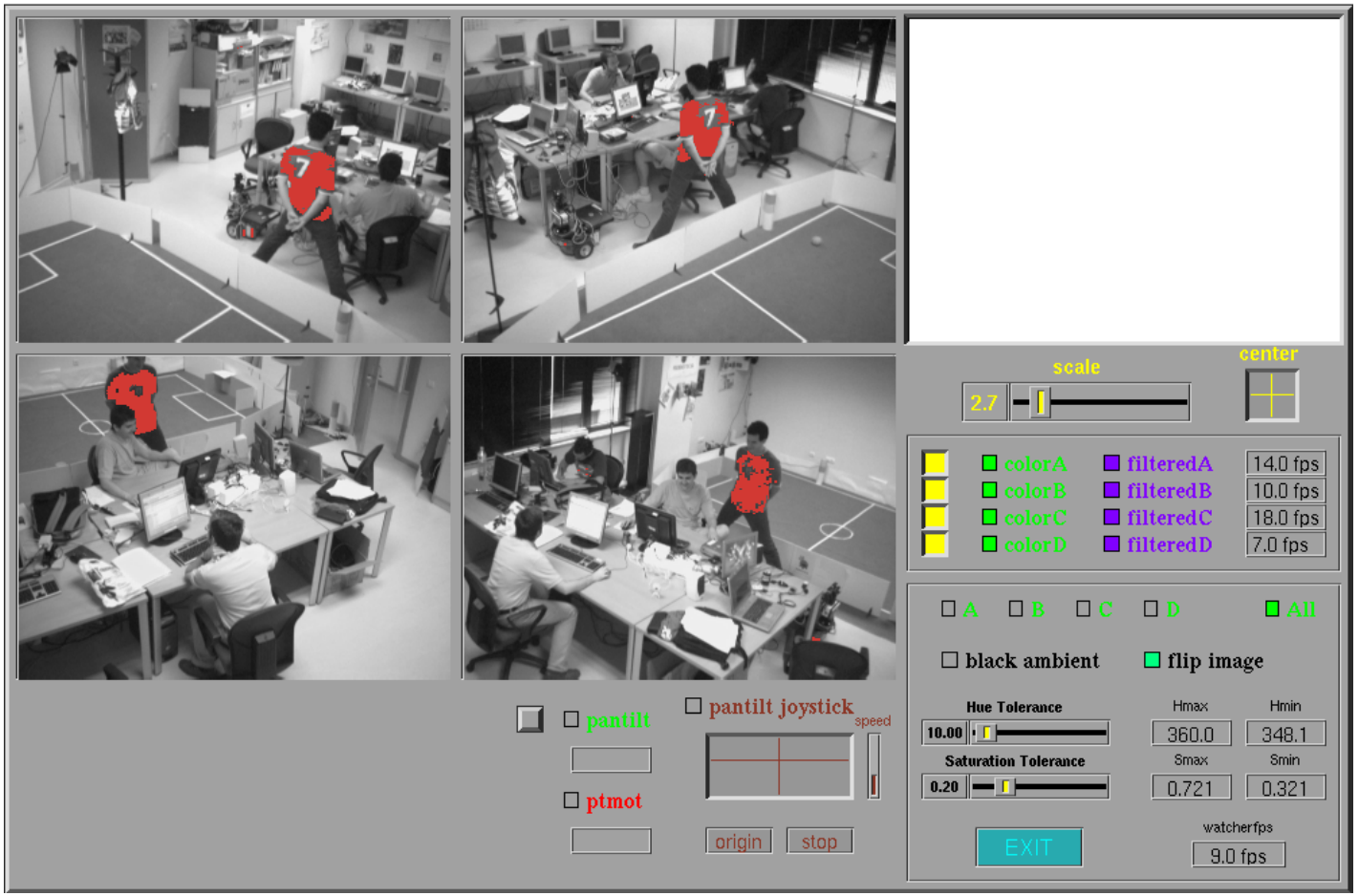




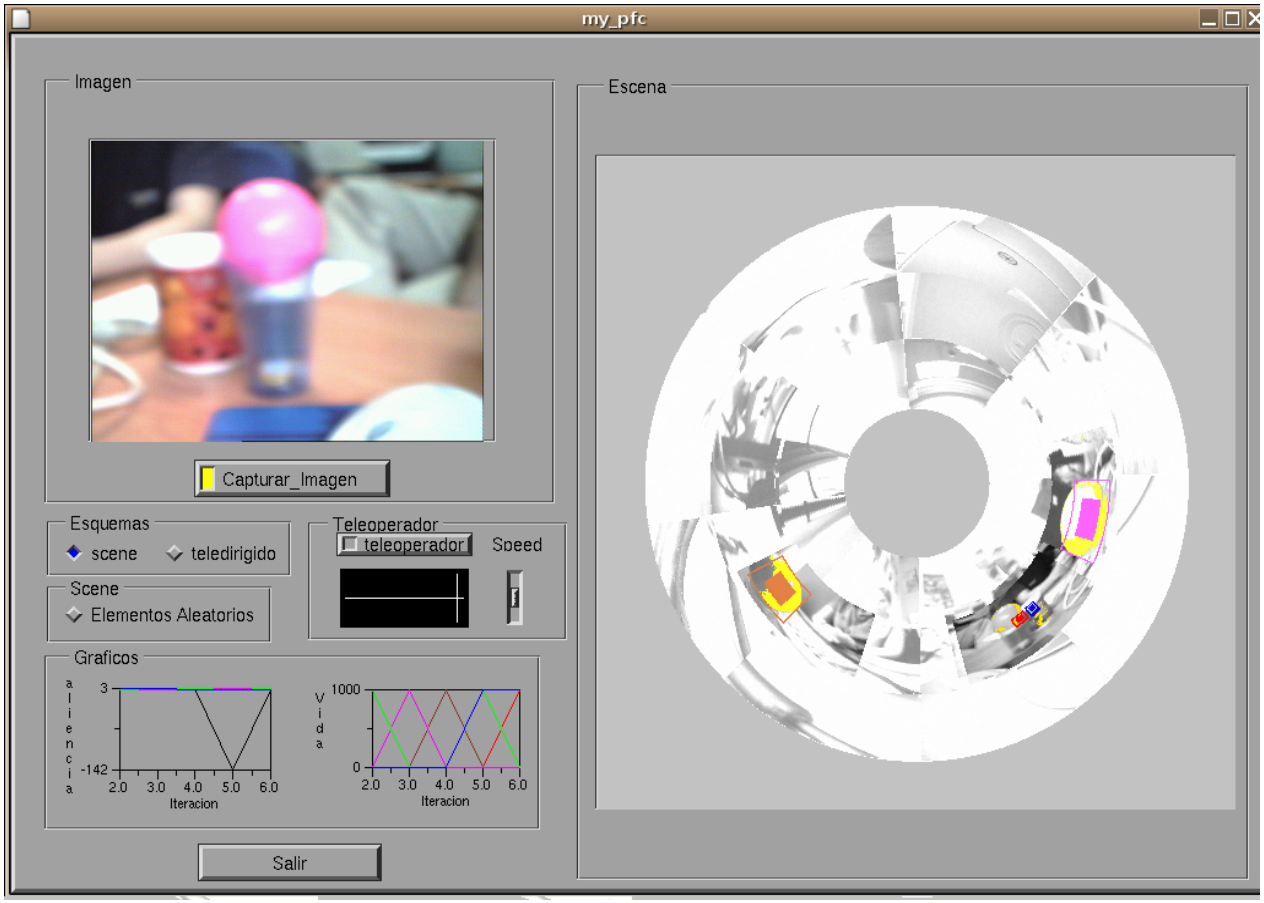
Visión avanzada: seguimiento 3D con filtro de partículas



Aplicación de seguridad con la misma tecnología



Visión avanzada: Atención visual



Conclusiones

- Generar un repertorio integrado de comportamientos es difícil
- La etología puede aportar modelos buenos para generar comportamientos mejores y más complejos en robots autónomos
- JDE: jerarquía dinámica de esquemas
- Implementada en software: JDE.c
- Comportamientos clásicos, taxias y visión "avanzada"

- Percibir a otro robot congénere
- Secuencias flexibles de taxias