
Robótica de Servicio

José María Cañas

josemaria.plaza@urjc.es



Grado Ingeniería Robótica Software, Curso 2023-2024

Introducción a la Robótica de Servicio

Contenidos

- Algunas definiciones
- Aplicaciones de los robots de servicio
- Programando la inteligencia de un robot de servicio

Algunas definiciones

- Robots

RAE : Máquina o ingenio electrónico programable que es capaz de manipular objetos y realizar diversas operaciones.

JoseMaría : Sistema informático con sensores, actuadores y computador(es), hay que programarlo para que consiga sus objetivos (realice su tarea) y sea sensible a la situación de su *entorno*. En su *software* reside su inteligencia.

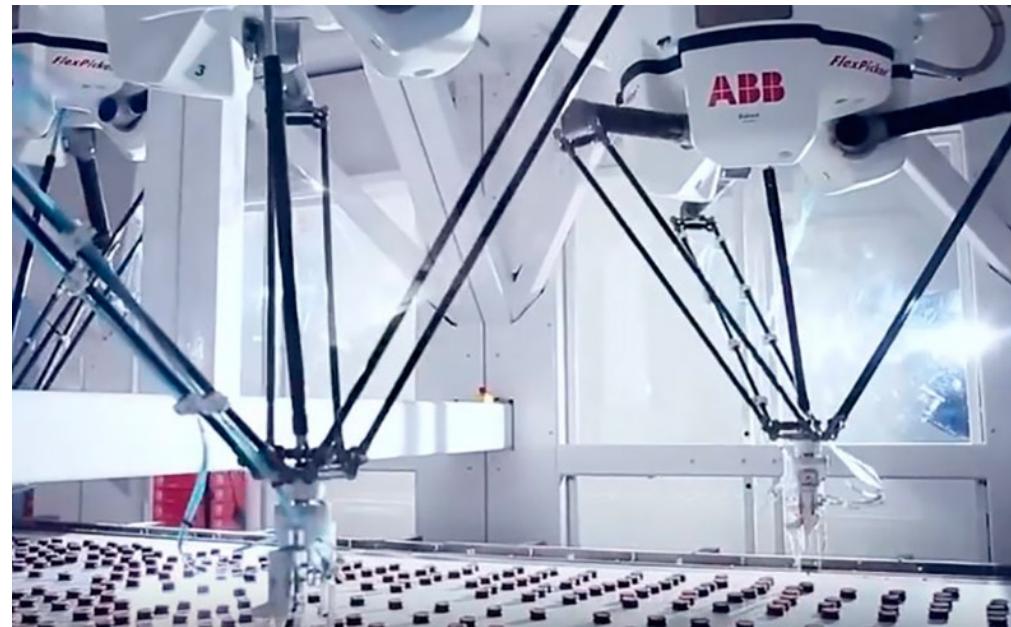
- ¿Qué tareas? 4D: *Dull, Dirty, Dangerous, Delicate*
- Robots industriales
- Robots de servicio

- Utilidad: aumentan productividad, seguridad, disminuyen costes
- Aplicaciones robóticas: sueños → investigación → mercado
- Requisitos: agilidad, **robustez**, precio
- El software es fundamental

Robots industriales



- Brazos robotizados y manipuladores, no deambulan
- > 3 grados de libertad
- Entorno controlado (con la excepción de los cobots), factorías
- Aplicaciones: *pick & place*, soldadura, ensamblado de piezas, pintado, manipulación...
- Efectores dependientes de la aplicación: pinza, ventosa, soldador...



- Se utilizan en muchas industrias: automoción, electrónica, química...
- Proveedores asentados: KUKA, ABB, FANUC, Festo...
- Mucho control en posición, planificación de trayectorias, cinemática inversa y directa... (+visión)

Robots de servicio



- ISO 8373: a robot “that performs useful tasks for humans or equipment *excluding industrial automation applications*”
- No-industriales
- Entorno no tan controlado
- ≤ 3 grados de libertad

- Aplicaciones muy heterogéneas
- Hardware y morfologías muy variados
- Inteligencia (**autonomía**): cumple objetivos + reactividad razonable
- Diferentes grados de autonomía (teleoperados también son útiles)
- Cuanta más autonomía más potencial

“the ability of autonomous robots to do jobs in unmanned environments makes them viable to perform tasks in a much wider range of industries” (David Inggs, CEO of Rocos)
- Mercado inmaduro: muchas start-ups, más que proveedores asentados
- Mercado en crecimiento

- Escenarios no controlados → se necesita mayor reactividad
- Escenarios variados (mucho más que industriales), sin adaptar
→ percepción más difícil, robustez más difícil

p.e. conducción autónoma en lluvia, soleado, niebla, de noche, con pocos coches en carretera, con muchos, lentos, rápidos, muchos tipos de carretera, peatones, ciclistas, obras, asfalto, caminos...
- Más difíciles de programar que los robots industriales

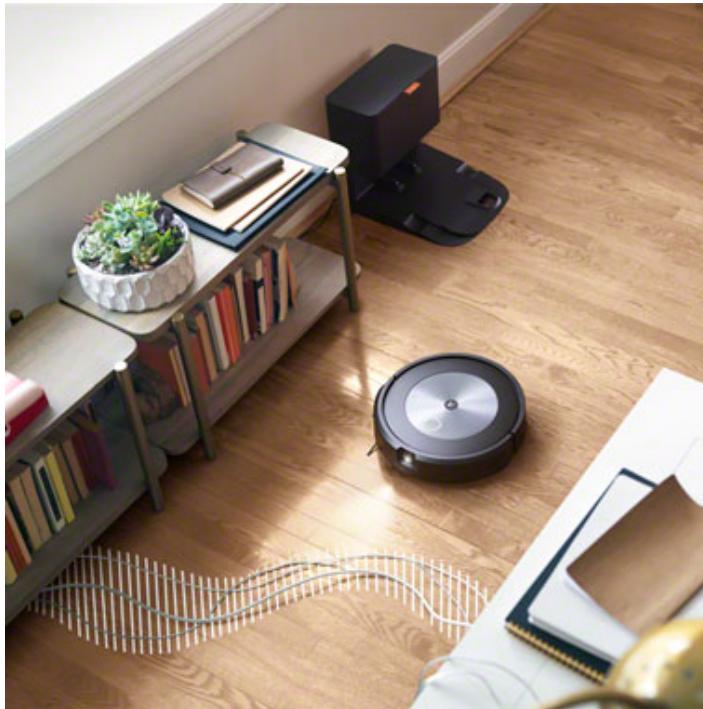
Aplicaciones de los robots de servicio

- Es móvil la frontera entre lo realizable actualmente con robots y lo que no. Con las mejoras tecnológicas, cada vez se pueden hacer más tareas con robots
- Enorme progreso en la última década
- Prototipo no es producto
- Hay muchas aplicaciones aún no resueltas (*¡muy difíciles!*): cuidado de personas, tareas domésticas
- Servicios profesionales
- *By 2021, the professional service robotic market is predicted to reach \$ 37 Billion. Businesses want to automate certain processes for safety, efficiency and productivity*

AGRICULTURE
CONSTRUCTION / DEMOLITION
CONSUMER
DEFENSE / SECURITY
ENERGY / MINING
HEALTH / MEDICAL
LOGISTICS / SUPPLY CHAIN
MANUFACTURING
PUBLIC SAFETY
RETAIL
ROBOTICS DEVELOPMENT
UTILITIES



Aplicaciones domésticas



- limpieza
- entretenimiento: juguetes, robótica educativa

Robots en medicina, asistencial y social



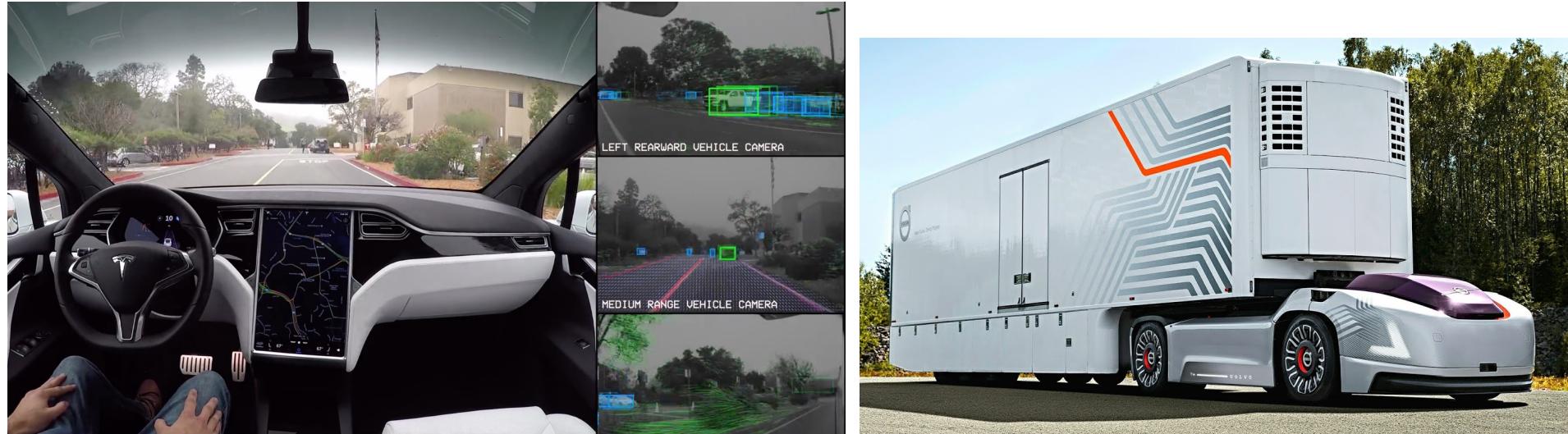
- de cirugía
- telepresencia
- exoesqueletos y rehabilitación
- reparto de material médico
- humanoides...

Robots en logística



- en almacenes: mover estanterías
- en cadenas de montaje: mover piezas
- entregas (última milla)

Conducción autónoma



- niveles de autonomía: desde ayuda a control total
- control de crucero, autoparking, atascos, autopistas, urbano...

Robótica de campo



- agricultura, minería, espacio, militares
- plantar, fumigar, cosechar, recolectar...

Inspección, mantenimiento y vigilancia



Security robots are the size of a person and they travel around corporate offices, equipped with heat sensors, facial detection and employee badge scanners. The robots rove autonomously, watching for signs of trouble, like an unauthorized person entering the building

Programando la inteligencia de un robot de servicio

- (hw) Computadores, sensores y actuadores específicos para la tarea concreta
Von Neumann (microcontroladores, CPUs), GPU, FPGAs
- (sw) Aplicación robótica = Percepción y toma de decisiones en el contexto de una aplicación concreta
 - distintos niveles de abstracción y frecuencias
- Requisitos software
 - Robustez es fundamental, aquí DL ha ayudado
 - Agilidad
- (BBS) *Brain + Body + Scenario*



Ingredientes, capacidades básicas

- Reactividad (control), razonamiento simbólico (planificación, deliberación), gestión de ejecución (hsm, bt...)
- Percepción, entender la escena, fusión sensorial, detección de objetos, visión artificial (2D/3D)...
- Navegación autónoma (mapas, localización, navegación local/global)
- *Human Robot Interaction* (visual, auditiva, botones, pantallas, NLP)
- Seguridad (es más importante en productos que en prototipos)

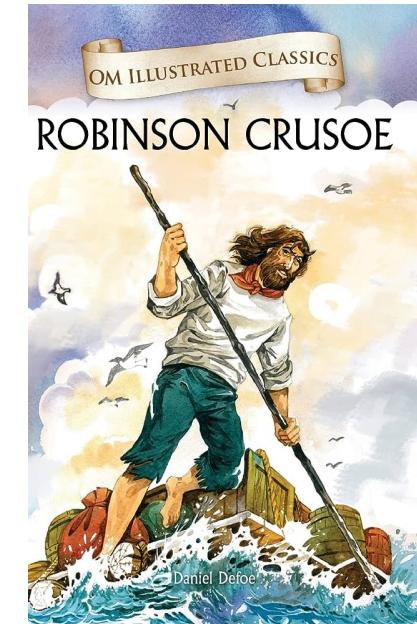
Integración

- Los sistemas robóticos son complejos, *integran* muchos componentes, subsistemas
- Desarrollo software: Robinson Crusoe vs Integrador
- Middleware robóticos: herramientas, drivers, bloques reutilizables de funcionalidad resuelta (bibliotecas, nodos, *stacks*...)
- *Stacks*: ROS2 Navigation stack, PX4, ArduPilot, MoveIt...
- Reutilización disminuye tiempos de desarrollo y aumenta calidad
- Hay un **corpus de software robótico** que hay que conocer.
OJO evoluciona, hay que estar al día

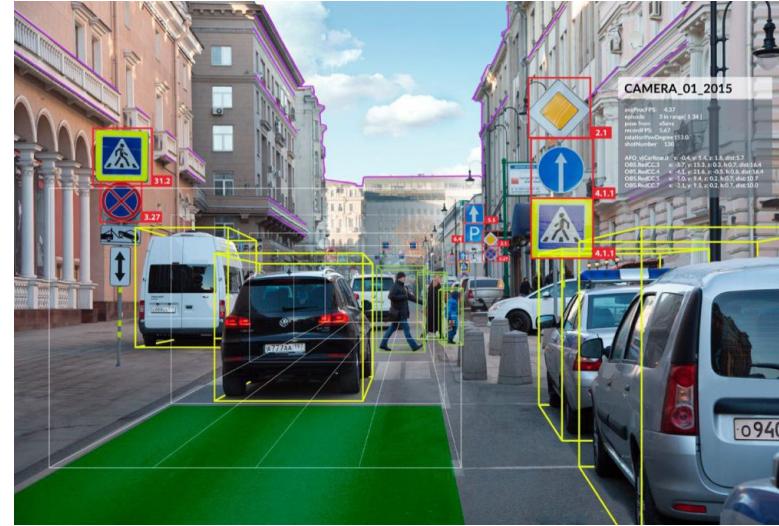
Habilidades para integrar Y crear código

- Integrar
 - requiere **conocer el software existente**, para reutilizarlo

p.e.: MoveIt para brazos, PX4 y MAVROS para drones, ROS2 Nav2 para navegación autónoma...
 - requiere **conocer los algoritmos** que hay dentro, para adaptarlos si es necesario
- Crear desde cero, diseñar un algoritmo y programarlo
- *Ambas son necesarias en un ingeniero robótico*
- Reutilizar → desarrollo en menos tiempo y mayor calidad sw porque las piezas las ha probado mucha gente



Algoritmos y aprendizaje automático

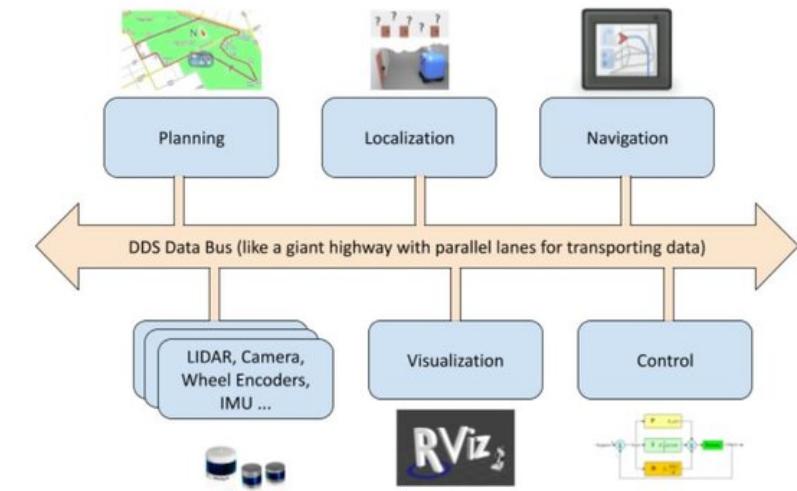


- Algoritmos → software (programación explícita)
- Aprendizaje automático:
 - No sólo en percepción, también en control y extremo a extremo
 - datos → software (deeplearning)
 - ensayos → software (aprendizaje por refuerzo)



Arquitecturas software

1. Secuencial Iterativa
2. Autómatas (HFSM) y árboles de comportamiento (BT)
3. Orquesta de nodos, distribuida
4. Híbridas de tres niveles
5. *DeepLearning*, indirecta
6. *Reinforcement Learning*, indirecta
7. *Reconfigurable Computing*: circuitos



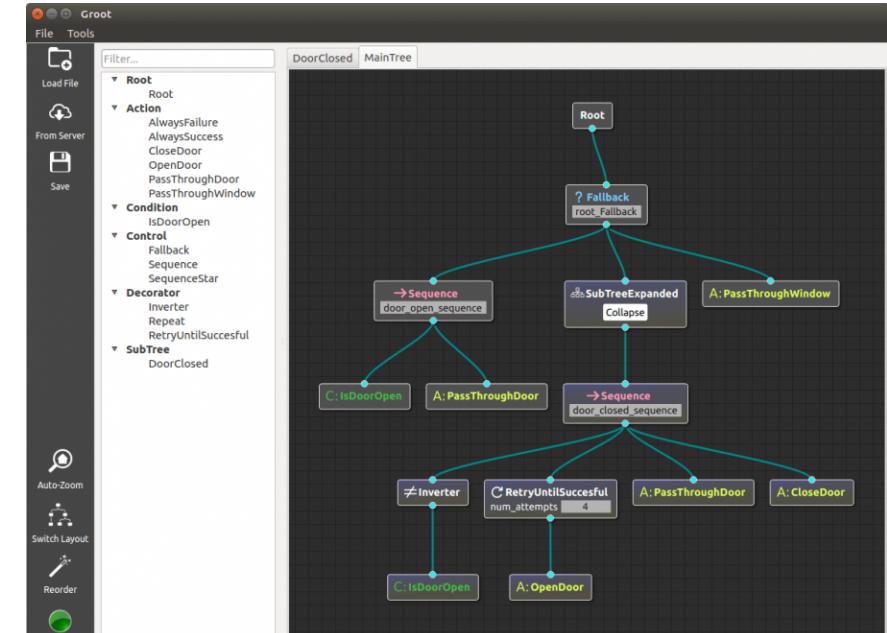
- ¿Cómo juntar todos los ingredientes/capacidades?
- Programación explícita vs indirecta

SECUENCIAL ITERATIVA

- Parte inicial con secuencia de instrucciones que se ejecutan 1 vez
- Parte iterativa en bucle infinito
- Microcontroladores
- Frecuencia de iteraciones es muy importante
- Cuadra con sistemas reactivos: cada iteración percepción y decisión

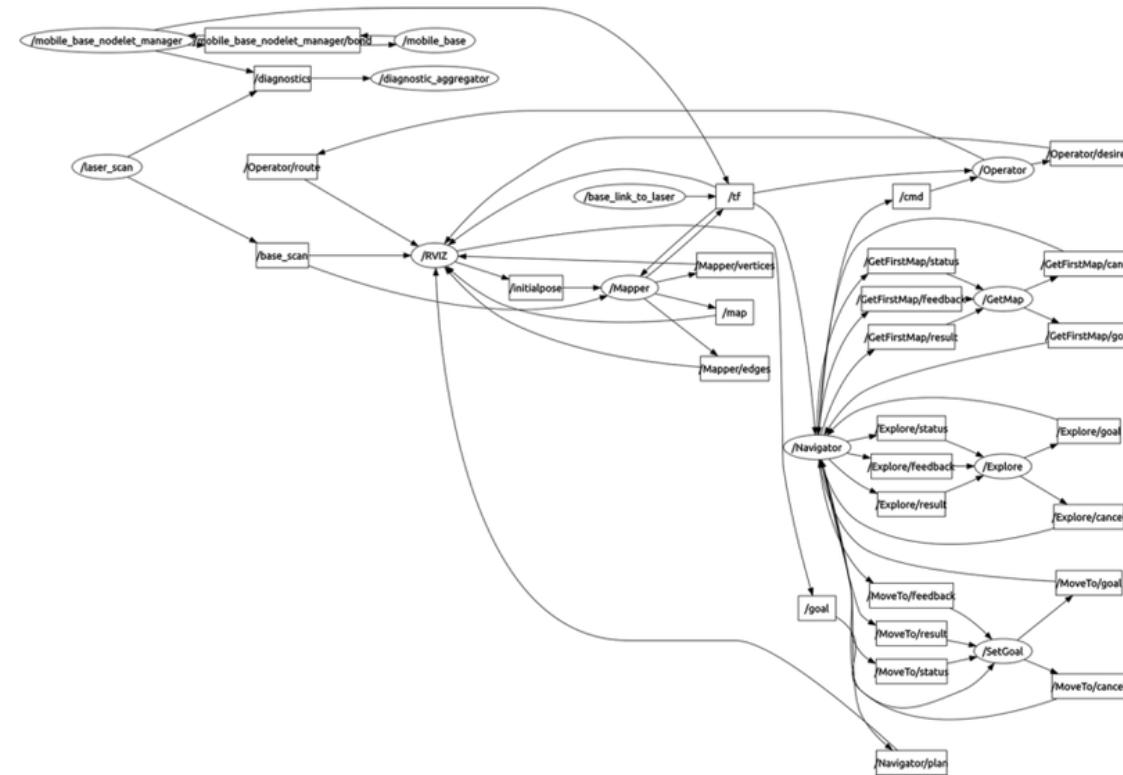
AUTOMÁTAS Y ÁRBOLES DE COMPORTAMIENTOS

- HFSM:
estados, transiciones,
niveles, precondiciones
- BT:
conectores secuencia, paralelismo...
tareas con resultado (éxito/fallo)
condiciones
reutilizables



- Control de la ejecución de partes
- Integran comportamientos variados, imposible con reactivos puros
- Ejemplo en videojuegos, inteligencia de los jugadores automáticos

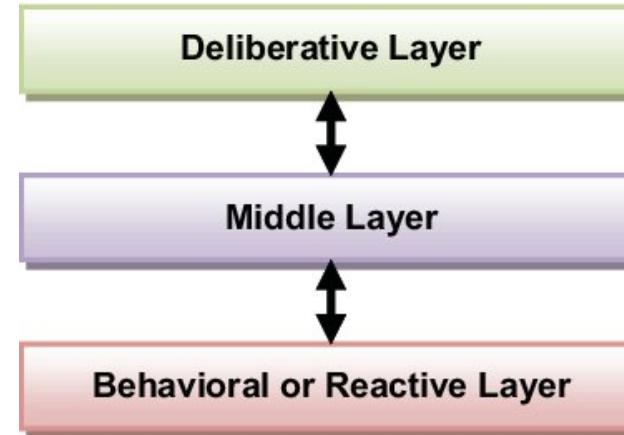
ORQUESTA DE NODOS



- Distribuida
- Nodos se intercambian información via ROS-topics, ROS-services...

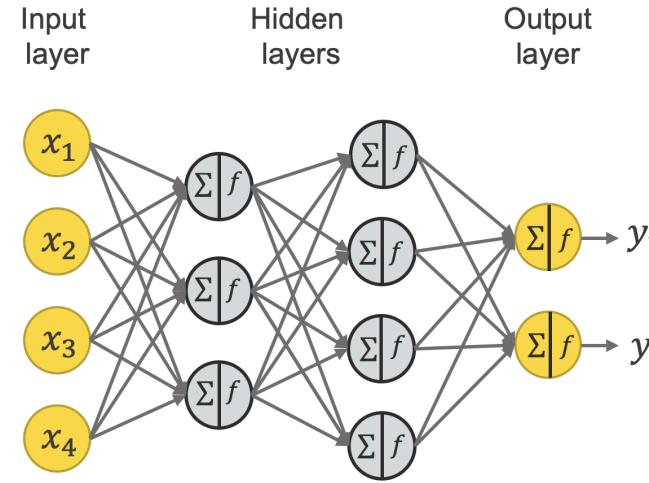


HÍBRIDAS DE 3 NIVELES



- Nivel deliberativo, razonamiento simbólico, planificación
- Nivel intermedio, gestión de ejecución, (des)activación unidades reactivas
- Nivel reactivo, controladores, alta frecuencia
- Ejemplo navegación autónoma:
plan global (ruta), secuencia subobjetivos, *obstacle avoidance*

REDES NEURONALES, *DeepLearning*



- Otra manera de organizar el cómputo: **modelo** = topología + pesos
- Aprendizaje supervisado, colección *enorme* de **datos**
- **Entrenamiento** (ajusta pesos) vs inferencia
- ¿En robots de servicio?: en percepción, en control y extremo a extremo
- Aporta robustez

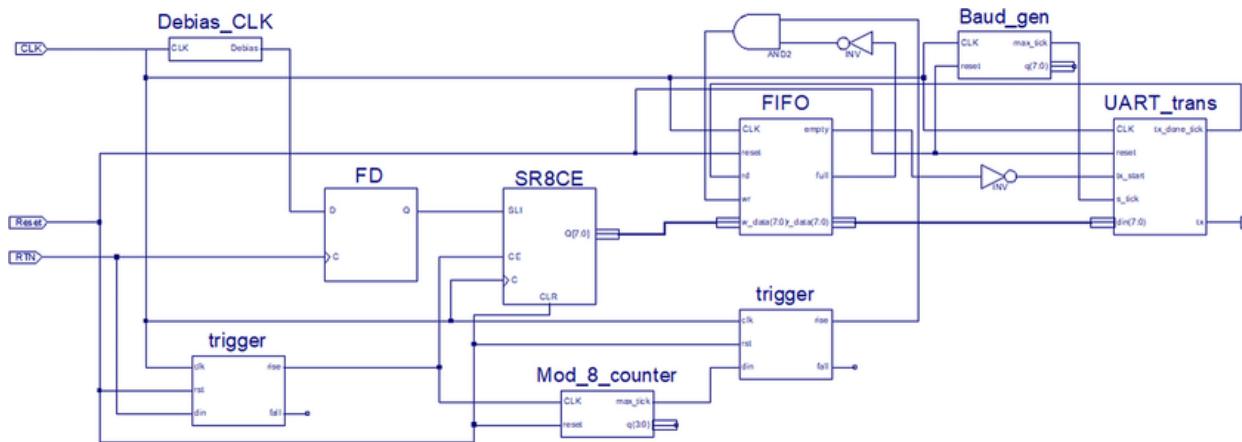


APRENDIZAJE POR REFUERZO, *Reinforcement Learning*

- Ensayo y error, muchos intentos
- Agente y ambiente
- Acción, estados, **recompensas**
- Q-Learning, DQN...
- ¿En robots de servicio?: control (p.e. aparcamiento)



COMPUTACIÓN RECONFIGURABLE



- Distribuida, paralelismo
- Bloques con entradas y salidas, se intercambian información vía cables
- FPGAs, ejecución muy rápida
- VisualCircuit, aproximación sw, cada bloque un proceso Python