

# Técnicas de los Sistemas Inteligentes

Curso 2017-18

Práctica1: Robótica.

Sesion1. Conceptos Básicos de Robótica

# Temporización Práctica1

- Plan Sesiones
  - *Sesión 1: 21, 22, 23 Febrero*
    - Introducción robótica. Introducción ROS.
  - *Sesión 2: 1, 2, 7 Marzo*
    - Programación en ROS. Ejemplos paquete TurtleSim
  - *Sesión 3: 8,9,13 Marzo*
    - Simulador 2D Stage. Ejemplo de control de robot. Introducción a Rviz
    - **Definición Práctica1.**
  - *Sesión 4: 14, 15, 20 Marzo*
    - Localización y mapeo. Visualización (rviz). Navegación
    - Configuración de arquitectura ROS para problemas de navegación.
    - Navegación global basada en A\* I
  - *Sesión 5: 21,22 Marzo, 4 Abril*
    - Navegación global basada en A\* II.
  - *Sesión 6: 5, 6, 11 Abril*
    - Seguimiento/Dudas. **Entrega P1: 14 Abril hasta las 14:00**

# Temporización Prácticas 2 y 3

- Plan Sesiones
  - *Sesión 7: 12, 13, 18 Abril:*
    - Planificación Clásica con PDDL
  - *Sesión 8 19,20, 25 Abril:*
    - *Seguimiento.*
  - *Sesión 9 26, 27 Abril, 2 Mayo*
    - *Seguimiento*
  - *Sesión10 3,4, 9 Mayo (Día de la Escuela 10 de Mayo)*
    - *Seminario3: Agentes en Videojuegos*
  - *Sesión11 11, 16, 17 Mayo*
    - Planificación HTN con HPDL. **Entrega P2: 21 de Mayo hasta las 14:00**
  - *Sesión12 18, 23, 24 Mayo*
    - *Seguimiento*
  - *Sesión13 25, 30 Mayo*
    - *Seguimiento.* **Entrega P3: 3 de Junio hasta las 14**

# Práctica 1 :Objetivo

- Resolver problemas de navegación de robots (basándonos en A\*):
  - Capaz de moverse en un entorno simulado.
  - Reconocimiento del mapa del entorno de un robot.
  - Planificar rutas seguras entre las ubicaciones y navegar evitando obstáculos siguiendo las rutas planificadas
- Conocer terminología básica en robótica
- Conocer los problemas básicos en robótica y qué técnicas se emplean para su solución
- Conocer el entorno ROS (Robotic Operating System)
- Controlar robots móviles resolviendo algunos de los problemas básicos de la robótica
  - En un entorno de simulación de robots: Stage
- Escribir programas en C++, basados en ROS, que implementen las soluciones.

# Bibliografía para esta práctica

- **Capítulo 25 del libro.** *Artificial Intelligence: A Modern Approach: Stuart Russell and Peter Norvig*, (Prentice Hall, Englewood Cliffs, NJ, 1995);. Elsevier, 1996. (En biblioteca)
- Fernández, Enrique, Luis Sánchez Crespo, Anil Mahtani, and Aaron Martinez. *Learning ROS for Robotics Programming - Second Edition*. 2nd ed. Packt Publishing, 2015.
  - (acceso biblioteca electrónica  
<http://proquest.safaribooksonline.com/9781783987580>.)
- Goebel, R. Patrick. *ROS By Example INDIGO - Volume 1*, 2015 (En biblioteca)
- Robin R. Murphy, Introduction to AI Robotics, The MIT Press 2000 (En biblioteca)
- Wiki de ROS Documentation <http://www.ros.org/>

# Sesion 1

- Seminario
  - Introducción
    - Definiciones básicas
    - Categorías de robots
  - Hardware de robots
    - Arquitecturas de agentes
    - Sensores
    - Tipos de sensores
    - Detectores de rango (range finders)
    - Sensores de ubicación
    - Sensores propioceptivos
    - Efectores
    - Grados de libertad en Manipuladores (Espacio tridimensional)
  - Los problemas fundamentales en robótica
- Sesión Práctica
  - Introducción a ROS.

# Introducción

- Definiciones básicas
  - Robot
  - Sensor
  - Efector
  - Actuador
- Categorías de robots
  - Manipulador o brazo-robot
  - Robot móvil
  - Manipulador móvil

# Definiciones básicas

- Robot
  - Agente físico que realiza tareas mediante la interacción con el mundo físico: percepción y manipulación del mundo físico.





# Definiciones básicas (2)

- **Sensor**

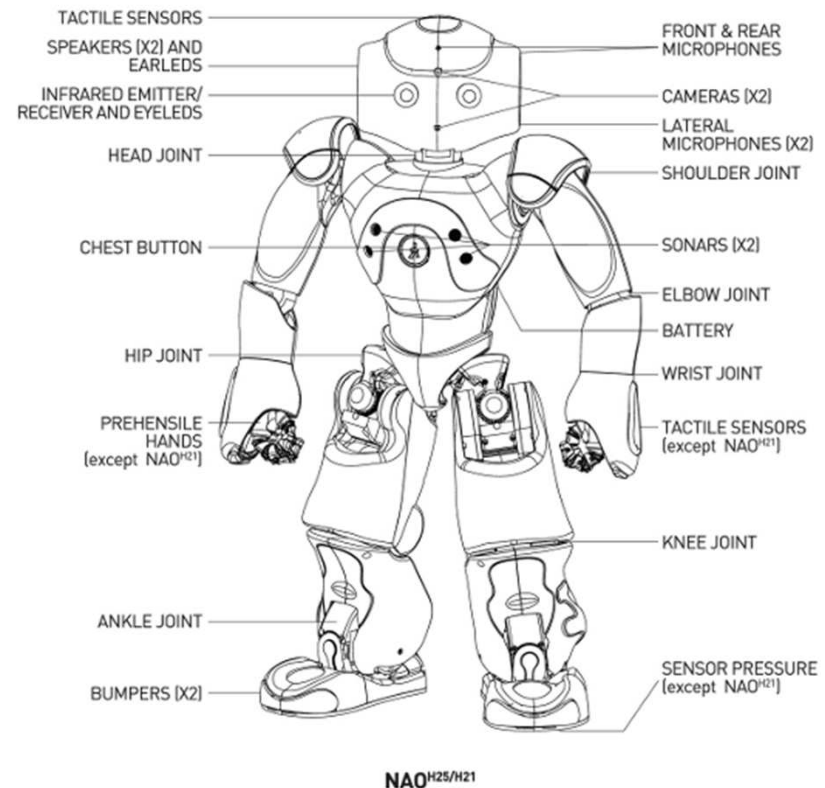
- Dispositivo que permite a un robot percibir el entorno. Un robot está dotado de varios conjuntos de sensores
  - Cámaras/láseres/infrarrojos/kinects para medir el entorno
  - Giróscopos o acelerómetros para medir el movimiento propio del robot

- **Efector**

- Dispositivo para manipular o modificar su entorno
  - Piernas, ruedas, articulaciones, pinzas,...
- Propósito: hacer efectivas fuerzas físicas en el entorno.

- **Actuador**

- Distinguirlo del efector: es la línea de control que comunica una orden al efector.
- Aunque no es la única definición



# Categorías de robots



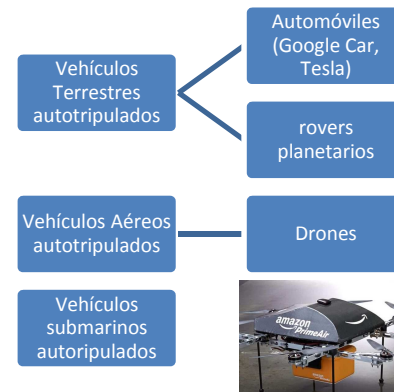
## Manipulador o brazo-robot

- Anclados físicamente en su espacio de trabajo. Movimiento involucra el control de una cadena de articulaciones controlables, habilitándolos a situar sus efectores en cualquier posición accesible desde el espacio de trabajo.



## Robot móvil

- Se mueven en su entorno usando ruedas, piernas o mecanismos similares. Distribuyen comida en hospitales, Mueven containers en muelles de carga.



## Manipulador móvil

- Humanoides /antropomorfos



NAO

# Categorías de robots



## Robots sociales.

- Interaccionan y se comunican con humanos u otros robots siguiendo reglas y comportamientos sociales

Kismet

de museo ...

Jibo



<http://www.myjibo.com/>

Care-O-bot

<http://wiki.ros.org/Robots/Care-O-bot>

Pepper

<https://www.alde.softbankrobotics.com/en/cool-robots/pepper/find-out-more-about-pepper>



## Categorías especiales:

- Investigación: robots no comerciales para experimentar en laboratorios de universidades

PR2



NAO



Turtlebot

# Robots educación, lúdicos, semi-profesional

## Lego MindStorms

- <http://www.lego.com/es-ar/mindstorms>

## MakeBlocks

- <http://www.makeblock.es/>

## Robots Arduino

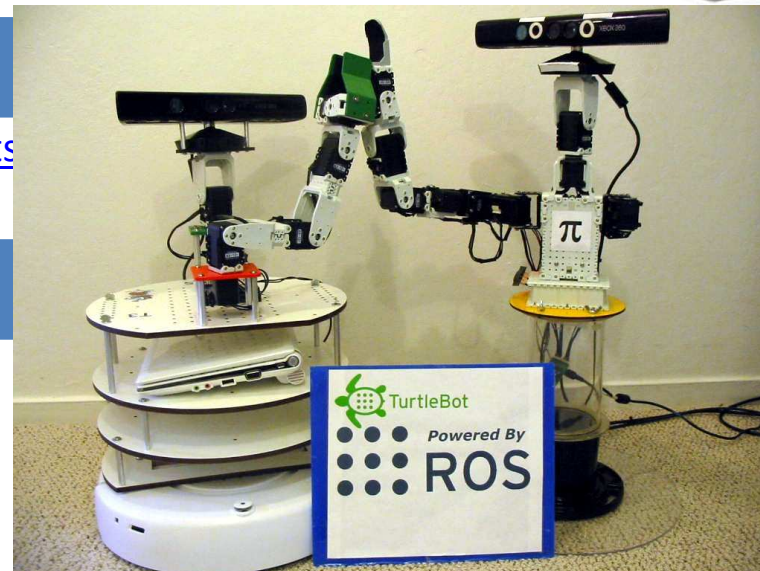
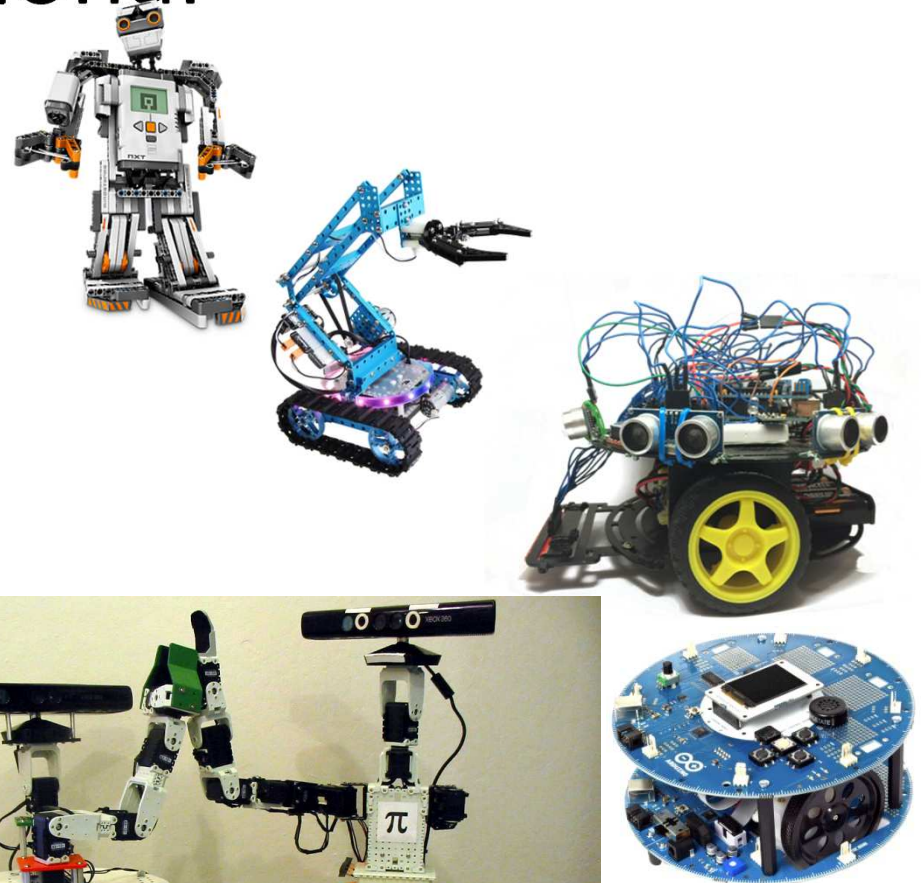
- <http://arduino.cc/en/main/robot>

## Robots Raspberry

- <http://www.raspberrypi.org/tag/robots>
- <http://wiki.ros.org/ROSBerryPi>

## PI Robot

- <http://www.pirobot.org/wordpress/>

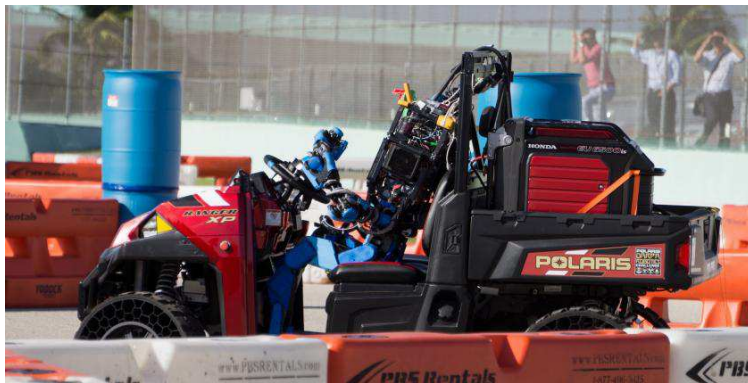




# Competiciones de Robótica

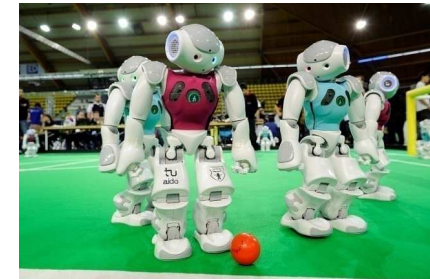
- **DARPA ROBOTICS CHALLENGE**
- Desarrollar robots semi-autónomos capaces de llevar a cabo tareas complejas en ambientes agresivos (como operaciones de rescate en catástrofes).
- No hay previstas más ediciones a partir de 2015.

<http://www.theroboticschallenge.org/>



## ROBOCUP

Promover investigación en robótica e IA.  
Desafíos complejos pero atractivos al público general. Iniciada en 1997.



<http://www.robocup.org/>

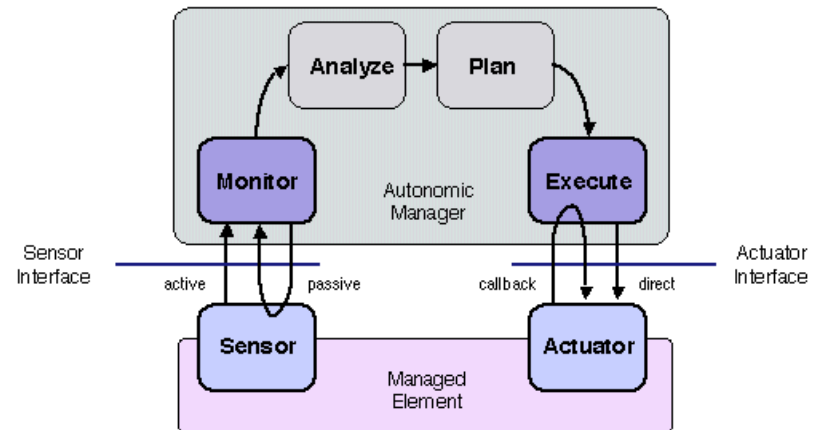


# **SENSORES**

# Arquitecturas de agentes

1. Percibir con sensores
2. Procesar con programas
3. Actuar con efectores

- El éxito de un robot real depende mucho de sensores y efectores.
- Es el principal obstáculo para la adopción generalizada de robots.



# Sensores

- Sensor
  - Interfaz perceptual entre el robot y su entorno
- Sensor activo (p.ej: sonar, láser)
  - envía energía al entorno que se refleja y vuelve al sensor
- Sensor pasivo (p.ej: cámara)
  - captura señales generadas por fuentes externas
- Los sensores activos facilitan más información que los pasivos
  - A costa de más energía y peligro de interferencias



# Tipos de sensores

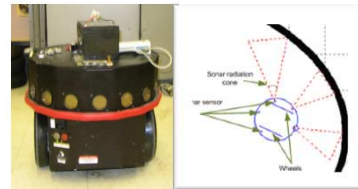
- clasificados en función de qué detectan
  - Sensores de rango: detectan objetos y su distancia en el entorno.
  - Sensores de ubicación: detectan la posición o localización del robot
  - Sensores propioceptivos: informan sobre la configuración interna del robot

# Detectores de rango (range finders)

Sensores que miden la distancia (range) a objetos cercanos

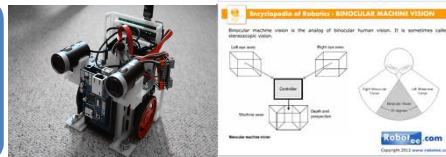
Características: rango de medida, campo de visión y fiabilidad de la medida.

Sonars



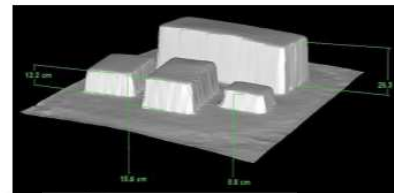
Rango y campo de visión pequeños. Sujeto a ruido. Un amplio campo de visión = Varios sensores sonar distribuidos

Camaras (estero visión)



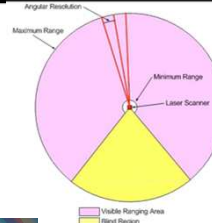
Actualmente usada raramente por no ser fiable.

Cámaras de tiempo de vuelo (time of flight cameras)



Usan láser, pero para rangos pequeños. Más fiable que sónar.

LIDAR (Láser)  
Light detection  
and ranging



Obtienen mayores rangos que las cámaras de tiempo de vuelo y funcionan mejor en luz natural. Muy caros todavía.

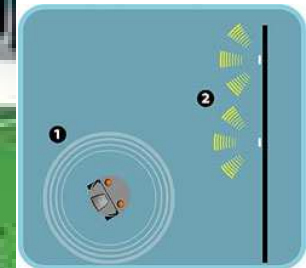
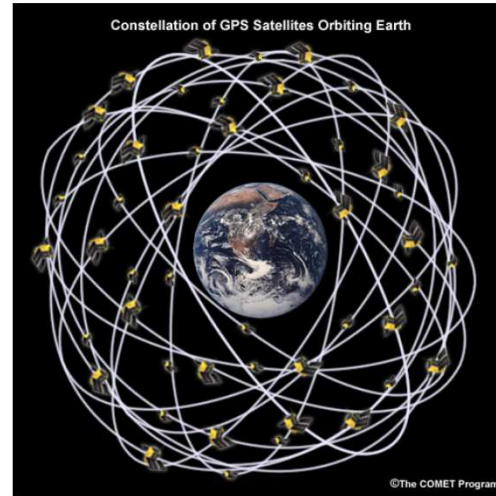
Kinect



Sensor asequible. Captura 3D. Un lidar es más caro y captura 2D. Ok en interiores, mal en exterior

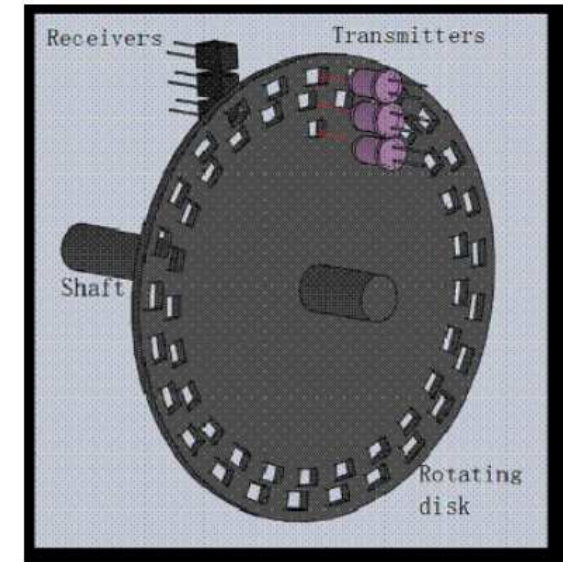
# Sensores de ubicación

- Informan sobre la posición del robot.
- el problema de localización no es trivial.
  - En el interior muchos algoritmos parten de sensores de rango como un componente primario para determinar la localización
    - Usando algoritmos de localización.
- GPS (bueno para exterior, malo para interior)
  - Ubicación absoluta por :Triangulación de señales de múltiples satélites.
    - Error de pocos metros
  - GPS Diferencial
    - Involucra un segundo receptor en tierra, facilitando precisión milimétrica.
- GPS no funciona en el interior ni en entornos submarinos
  - Uso de balizas de interior (p.ej.: landmarks en robocup).
  - Uso de repetidores wifi
  - Balizas sonar submarinas.



# Sensores propioceptivos

- Informan al robot de su propio movimiento
- **Encoders:** cuentan las revoluciones de un motor en pequeños incrementos.
  - Manipuladores: conocer la posición de una articulación
  - Robots móviles: revoluciones de una rueda

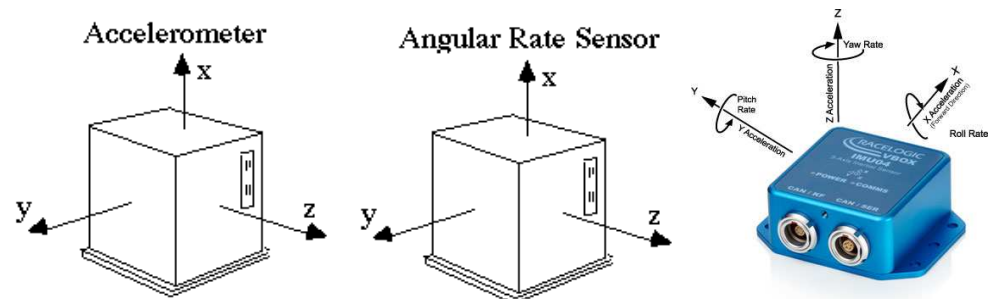


# Odometría:

- Concepto esencial en robot móviles.
  - medida o estimación de la posición de vehículos con ruedas durante la navegación
    - hodos = viaje, trayecto
    - metron = medida
- Consiste en la **determinación** de la **Pose** de un robot
  - **Pose** = (*posicion\_cartesiana, orientacion*) =  $(x, y, z, \theta_x, \theta_y, \theta_z)$
  - Odometría basada en encoders no es precisa, solo fiable para distancias cortas
    - En la realidad, la odometría siempre está afectada de incertidumbre.

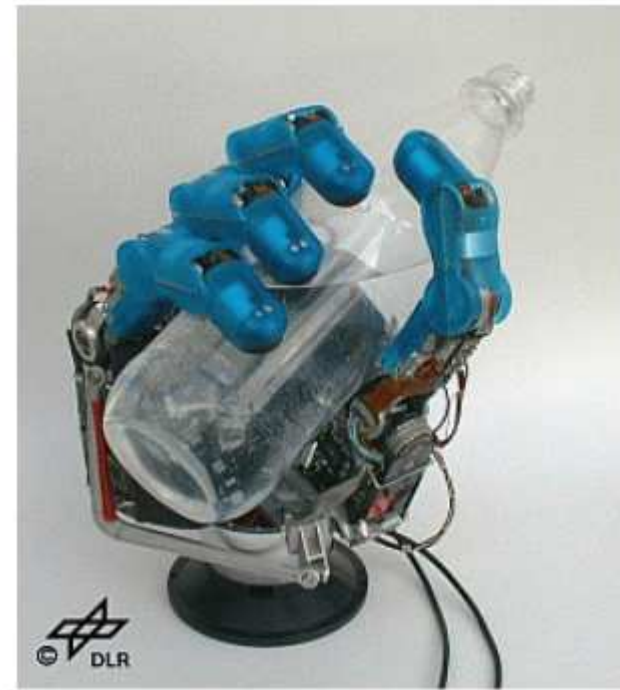
# Incertidumbre en odometría

- La incertidumbre siempre está presente en la medida de posiciones
  - Encoders con poca resolución.
  - Patinaje de ruedas (en robots móviles)
  - Fuerzas externas en vehículos terrestres
  - El viento en vehículos aéreos.
- Solución parcial: Sensores inerciales (giróscopos)
  - se basan en la resistencia de una masa al cambio de velocidad
  - Reducen incertidumbre.
- IMU (Inertial Measurement Unit): sensor inercial muy barato



# Sensores de fuerza o de torque

- Convierten torsión mecánica en señales eléctricas
- Indispensables para manejar objetos frágiles



# **EFFECTORES**

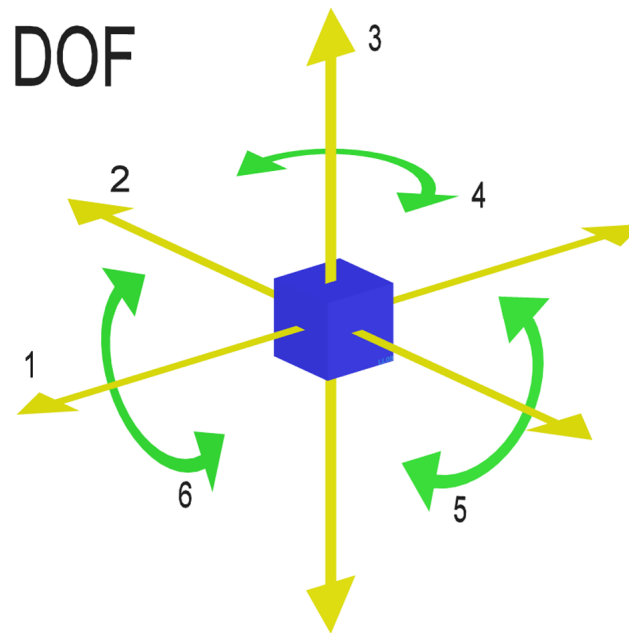


# Efectores

- Los efectores son el medio por el que los robots se mueven y cambian la forma de su cuerpo
- Hablaremos sobre movimiento y forma a un nivel lógico, no físico.

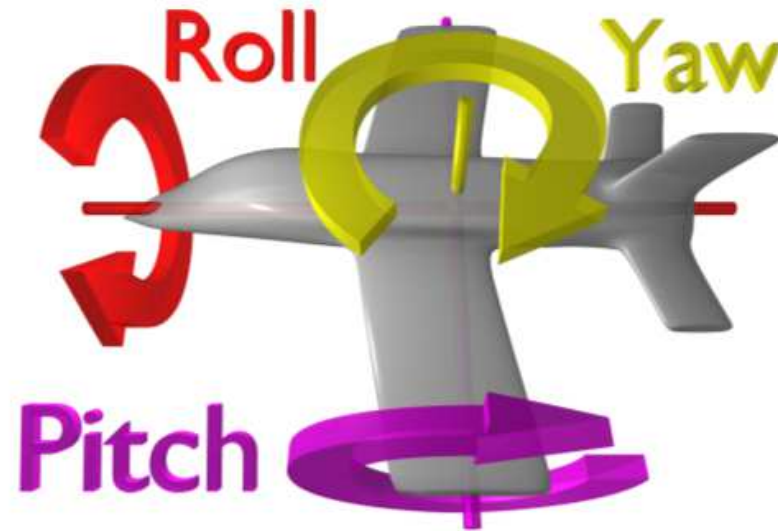
# Grado de Libertad (Degree of freedom: DOF).

- Cada una de las direcciones independientes (desplazamiento o rotación) en la que un robot (o uno de sus efectores) puede moverse.



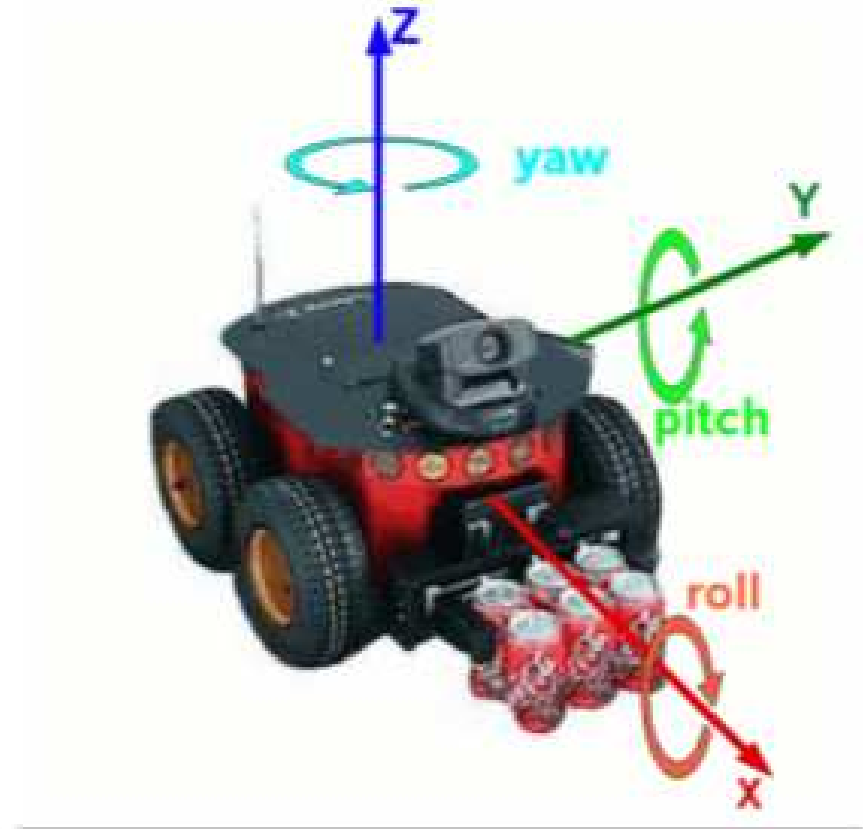
# Ejemplo robot móvil aéreo:

- Espacio tridimensional:  
6 grados de libertad
  - Posición espacio:  $x, y, z$
  - Orientación (términos náuticos):
    - roll (inclinación) giro en eje X
    - pitch(cabeceo) giro en eje Y
    - yaw(viraje): giro en eje Z



# Robot móvil estándar terrestre:

- Espacio bidimensional
- 3 grados de libertad
  - Situación en el espacio:  
 $x, y, z$  ( $z$  normalmente 0)
  - Orientación:  
 $pitch = roll = 0$
  - Pose =  $(x, y, 0, yaw)$
- Observar: este robot no puede inclinarse ni cabecear ni ascender



# Estado cinemático

- Los grados de libertad definen el estado cinemático o "pose" de un robot
  - Robot aéreo:  $(x, y, z, roll, pitch, yaw)$
  - Robot móvil:  $(x, y, z, yaw)$
- O de otra forma: los grados de libertad se corresponden con las coordenadas que necesitamos para definir el estado cinemático o **"pose"**

# Estado dinámico

- Incluye información sobre
  - posición + movimiento
- Posición:
  - pose =  $(x, y, z, yaw)$
- Una velocidad por cada parámetro del estado cinemático
  - velocidades =  $(v_x, v_y, v_z, \omega)$
  - $v_x, v_y, v_z$  son *velocidades lineales* (m/seg)
  - $\omega$  es *velocidad angular* (rad/seg ó °/seg)

# Grados de libertad en Robots móviles (Espacio bidimensional)

- No es necesario el mismo número de GDL que el número de elementos controlables
- Ejemplo: un coche estándar
  - Pose:  $(x, y, 0, yaw) = (x, y, yaw)$ :: 3 grados de libertad
  - Movimientos posibles:
    - Se mueve adelante/atrás, estableciendo una velocidad lineal positiva o negativa.
    - Puede girar un ángulo, estableciendo una velocidad angular durante un tiempo determinado.
- **Grados de libertad efectivos** = dimensiones de su estado cinemático
- **Grados de libertad controlables**: número de componentes sobre las que se puede actuar

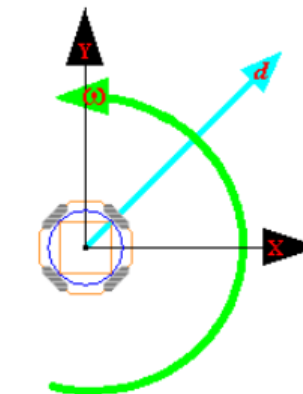
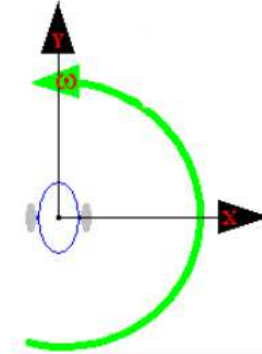
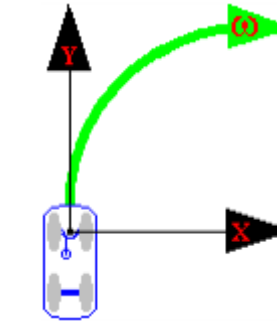
# Holonómico vs No Holonómico

- Robot no holonómico
  - Grados de libertad efectivos > Grados de libertad controlables
    - Coche
- Robot holonómico
  - Grados efectivos = Grados controlables
  - Es más fácil controlar un robot holonómico que un no holonómico
  - Un robot holonómico es más complejo mecánicamente
- En general
  - Manipuladores son holonómicos
  - Robots móviles son no-holonómicos



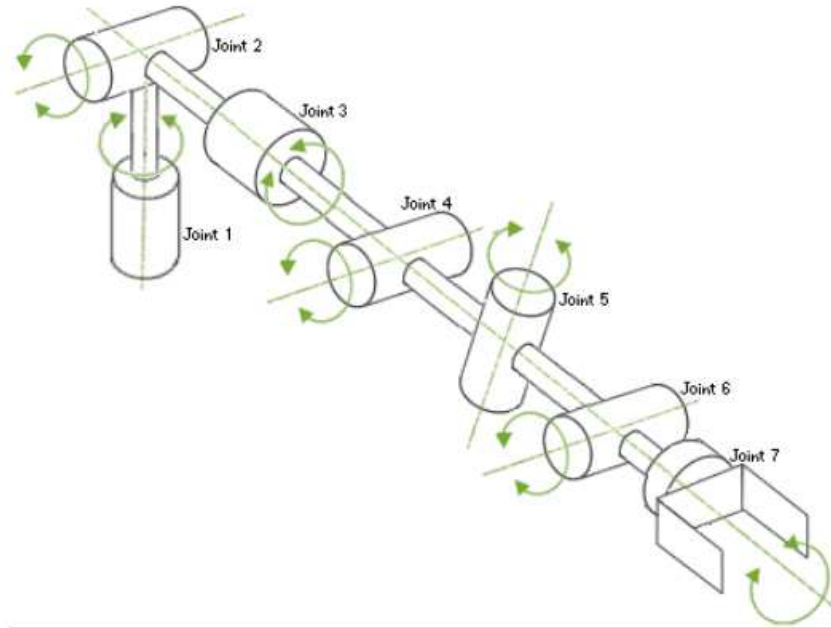
# Conducción en Robot móviles

- Conducción tipo-coche (car-like drive)
  - Dirección: hacia delante, hacia atrás.
  - Ángulo de conducción
- Conducción diferencial (differential drive)
  - Dos ruedas actuadas independientemente (o cadenas como los tanques)
  - Puede girar sobre sí mismo
- Conduccción sincronizada (omni drive)
  - Puede moverse en cualquier eje



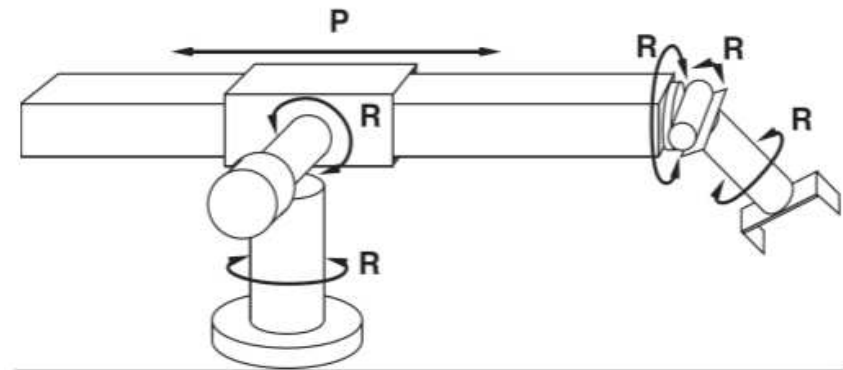
# Grados de libertad en Manipuladores (Espacio tridimensional)

- Los cuerpos no rígidos tienen articulaciones, cada una con uno o varios grados de libertad
  - Hombro humano: 2 grados de libertad
  - Muñeca humana: 3 grados de libertad



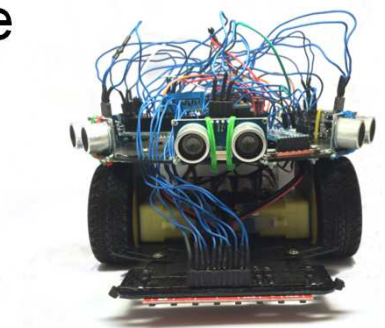
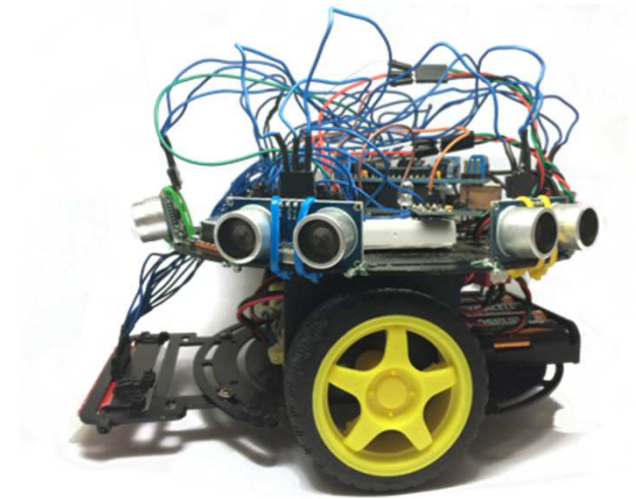
# Grados de libertad en Manipuladores

- Lo mismo puede ocurrir con articulaciones robóticas
  - El estado cinemático de la mano de un manipulador está definido por 6 componentes.
    - (x,y,z,roll,pitch, yaw)
    - Se requieren 6 grados de libertad para situar un objeto (una mano) en un punto con una orientación particular
  - Manipulador con 5 articulaciones circulares y una prismática
  - Manipuladores con grados de libertad extra son más fáciles de controlar que los que solo tienen el mínimo número.
    - En la industria son habituales los de 7 grados de libertad



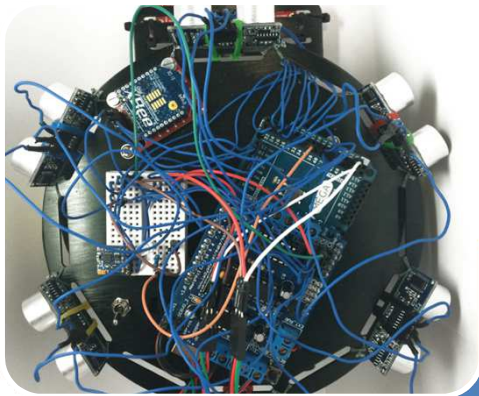
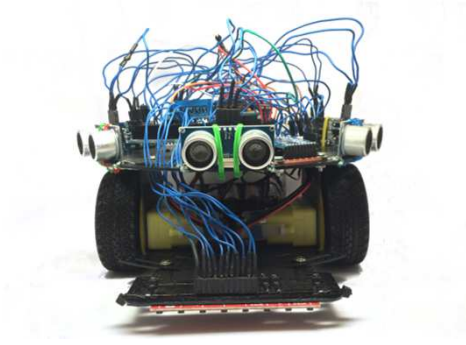
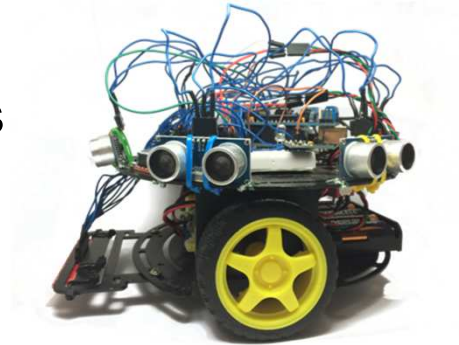
# Problemas básicos de la robótica (robot móvil)

- Percepción
- Movimiento y Control
- Localización
- Mapeo
- Navegación
  - Planificación local (reactiva) de movimientos
  - Planificación global (deliberativa) de movimientos
- Planificación de tareas

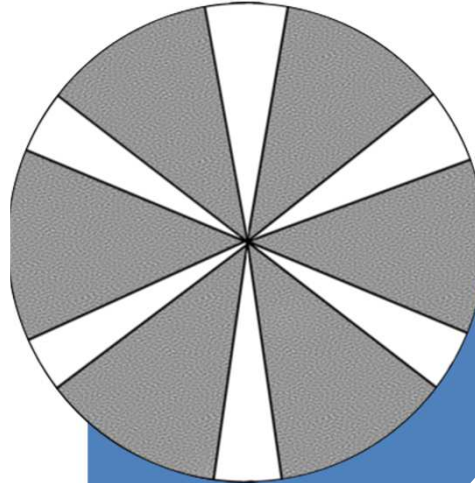


# Percepción

- Proceso por el que un robot transforma medidas externas de sensores en una representación interna del entorno



Robot con 6 sensores  
sónar distribuidos  
homogéneamente en un  
perímetro hexagonal



Representación Interna

- Ángulo de visión de  $15^\circ$
- Problemas para percibir superficies diagonales
- Lectura costosa en tiempo
- Array de valores representando  $360^\circ$  con valores no válidos

# Movimiento: Dinámica y control

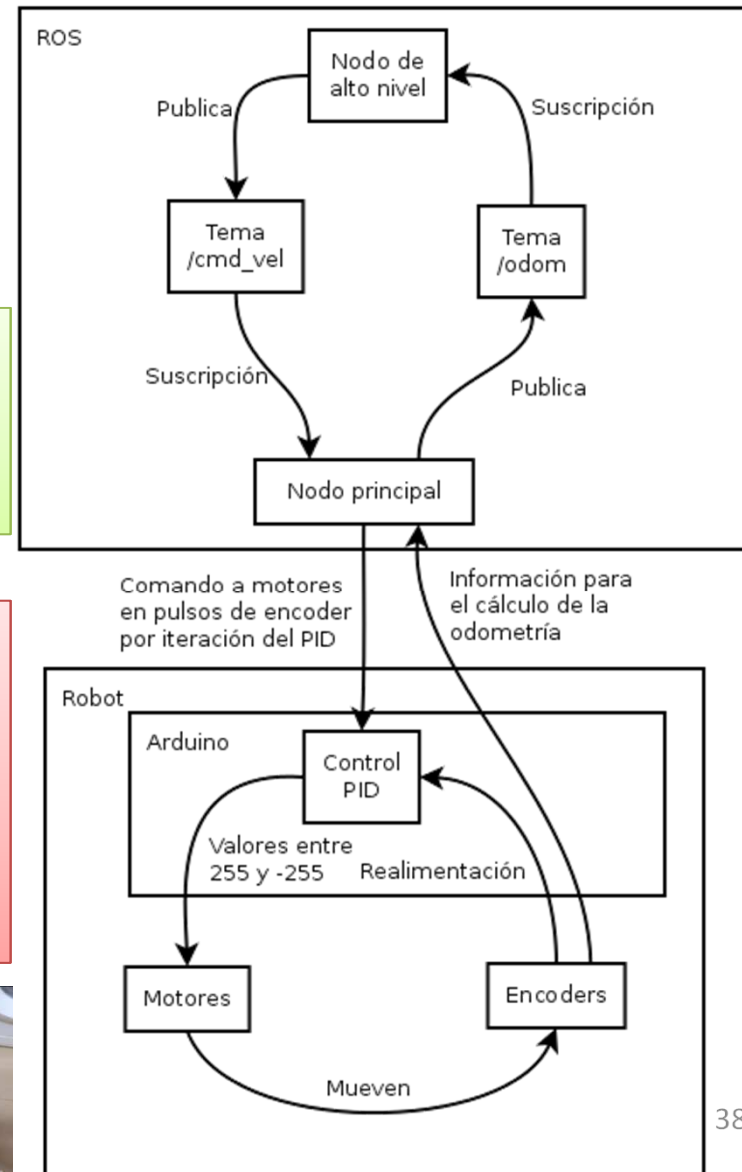
- El problema de determinar la velocidad o fuerza con que se tiene que mover un efector. Implica conocer aspectos físicos de movimiento e inercia (dinámica) y aspectos de control de sistemas para mantener la respuesta del robot en rangos aceptables.

## • ¿dónde implementar el comportamiento reactivo?

- Un nodo software recibe una velocidad lineal y una angular, y la convierte a una unidad que el robot pueda usar
- La odometría se calcula en el ordenador, combinando datos de los encoders y la brújula

## • En este ejemplo el control de bajo nivel del robot se implementa en el lado del robot

- se usan dos motores, dos encoders (posición) y una brújula (orientación)
- Resolución de los encoders: 20 pulsos por revolución. Unidireccionales
- Control PID en el robot



[Deambulación evitando obstáculos](#)

[Seguimiento de líneas en el suelo](#)





# Movimiento: Dinámica y control

## Distintas técnicas de control aplicables en robótica

### Control PID

- Técnica estándar (industria) para mantener un sistema continuo cerca de un valor de consigna.
- Ejemplo: mantener la velocidad cte. a 1m/sg.

### Control reactivo

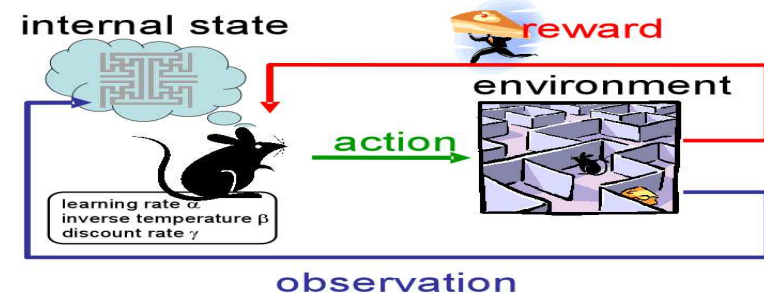
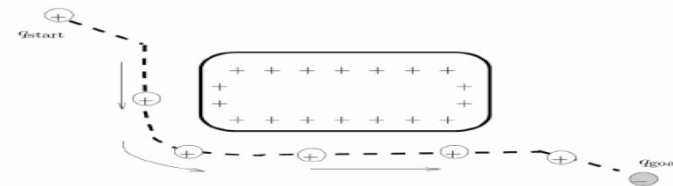
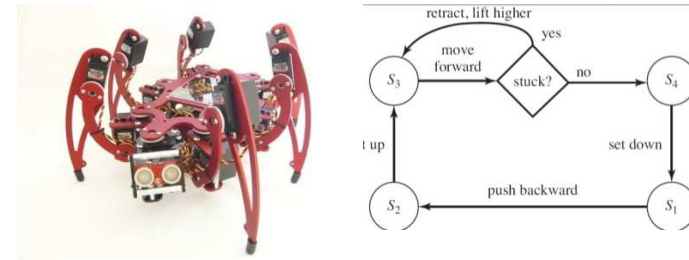
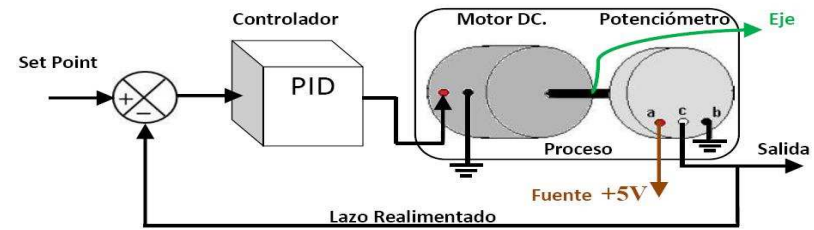
- Técnica basada en autómatas de estados (o en reglas) para controlar un sistema discreto.
- Ejemplo: Control de una pierna de un hexápodo.

### Control de campo de potencial

- Técnica usada para mantener un sistema en la consecución de un objetivo.
- Ejemplo: consigue llegar a la posición (x,y) .

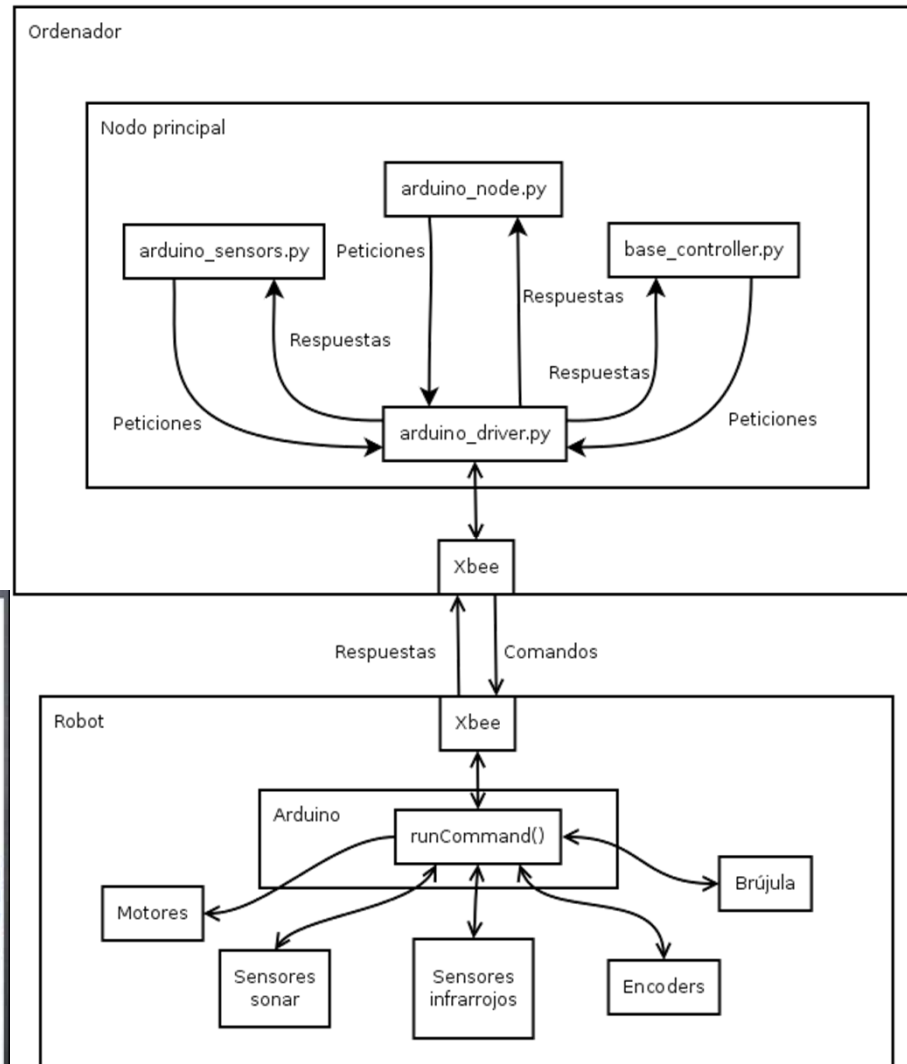
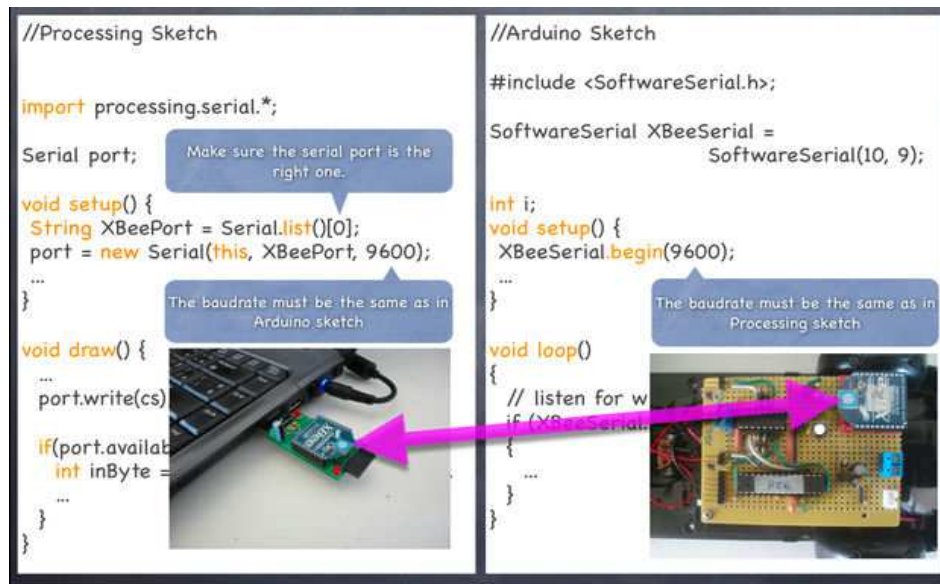
### Control por aprendizaje por refuerzo

- Técnica usada cuando el sistema es desestructurado y no hay un modelo previo.
- Ejemplo: encontrar las reglas de comportamiento para un robot bípedo.



# Comunicación Robot-Alto Nivel

- El control a bajo nivel de un robot **implica enfrentarse con problemas de comunicación**. Por ejemplo:
  - El ordenador envía comandos sencillos al robot. Por ejemplo "m 0.3 0.3"
  - El robot realiza la orden y devuelve los datos generados
  - En nuestro caso concreto, se usan dos módulos Xbee, que están configurados para crear un vínculo entre ellos
  - Además, el robot tiene un LED que proporciona feedback del punto de la ejecución en el que se encuentra
- La latencia de la comunicación influye en la percepción y el control de movimientos**



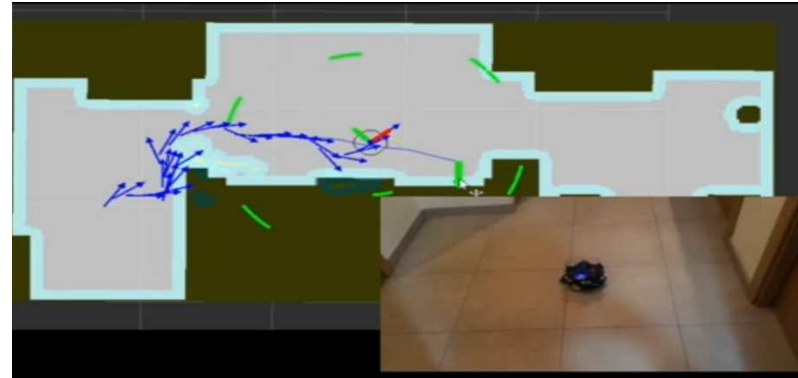


# Afortunadamente

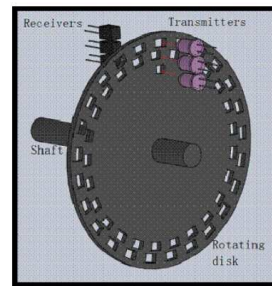
- Hoy en día, los problemas de percepción y movimiento (dinámica, control, comunicación) se resuelven de forma (casi) transparente con las actuales plataformas de robótica (como ROS, que conoceremos a continuación.
  - A no ser que queramos diseñar un robot desde cero, como ocurre en la robótica lúdica.
- El resto de problemas que veremos a continuación requieren en general altos recursos computacionales y sus soluciones se implementan en un ordenador conectado con un robot.

# Localización

- El problema de encontrar/determinar dónde están las cosas, incluido el robot mismo.
  - **Localización sin mapa:** usando la información propia de posición (odometría) que se calcula dentro del robot (sensores propioceptivos).
- **Sujeto a errores de estimación**
  - P.ej: odometría basada en encoders no es precisa, solo fiable para distancias cortas
  - En la realidad, la odometría siempre está afectada de incertidumbre.



Localización usando los datos de la odometría



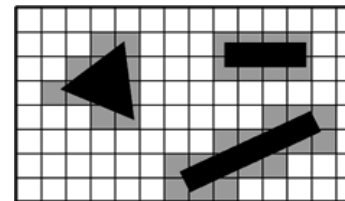
Encoder

# Localización con mapa conocido a priori

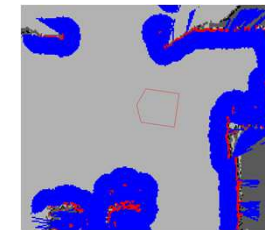
- **Requiere dos etapas:**

1. *Discretización del espacio*

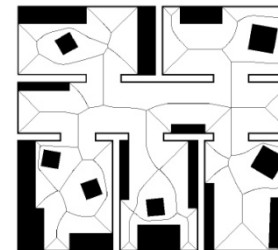
- Descomposición en celdas
  - **Matriz de ocupación** (Occupancy matrix): informa de celdas ocupadas, libres y desconocidas
  - **Mapas de coste** (Costmaps): proporciona información adicional a la matriz de ocupación sobre distancia a obstáculos.
- Esqueletización
  - **Grafo de Voronoi**: técnica de preprocesamiento sobre mapas para calcular posiciones equidistantes a obstáculos y facilitar navegación.



Matriz de ocupación



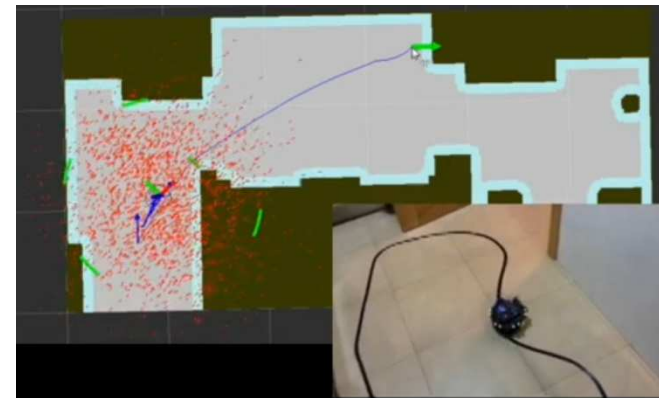
Mapa de Coste



Grafo de Voronoi

2. *Localización a partir de un mapa conocido del entorno.*

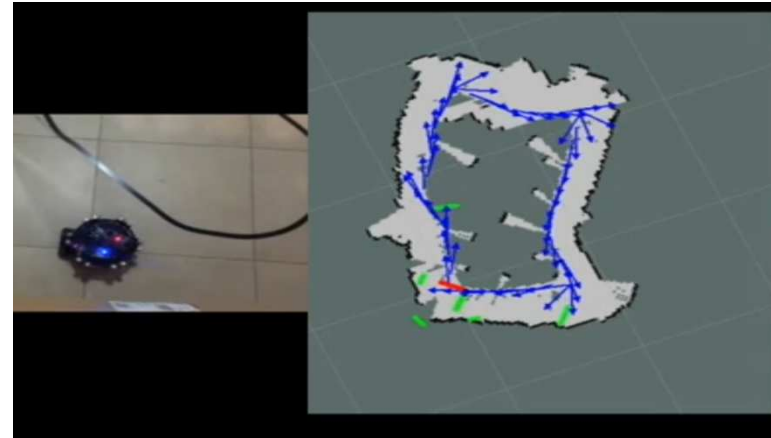
- **Localización con hitos** (landmarks): se conoce la posición exacta de ciertas balizas y la distancia a éstas.
- **Localización con sensores de rango**
  - **AMCL**: Adaptive Monte Carlo Localization, se estima la posición mediante técnicas estocásticas, analizando nubes de partículas simuladas.



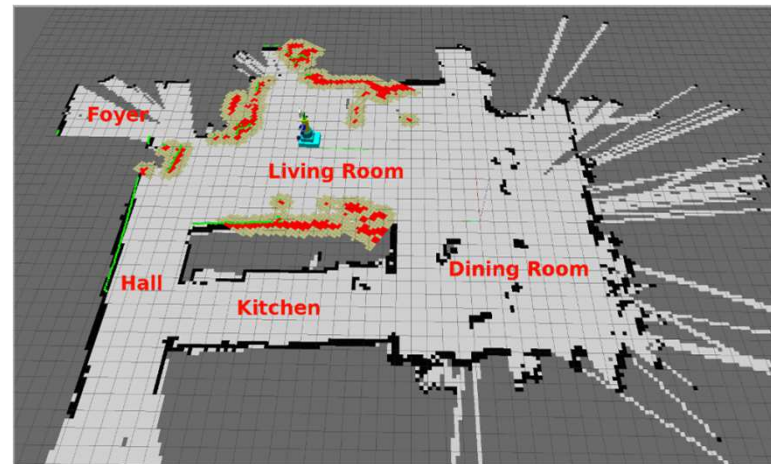
Localización usando amcl

# Localización y Mapeo

- Localización sin mapa inicial
  - Mapeo (Mapping)
    - Detección del mapa del entorno, con localización conocida (irreal excepto en simulación)
  - SLAM
    - Simultaneous localization and Mapping
  - Etiquetado de mapas

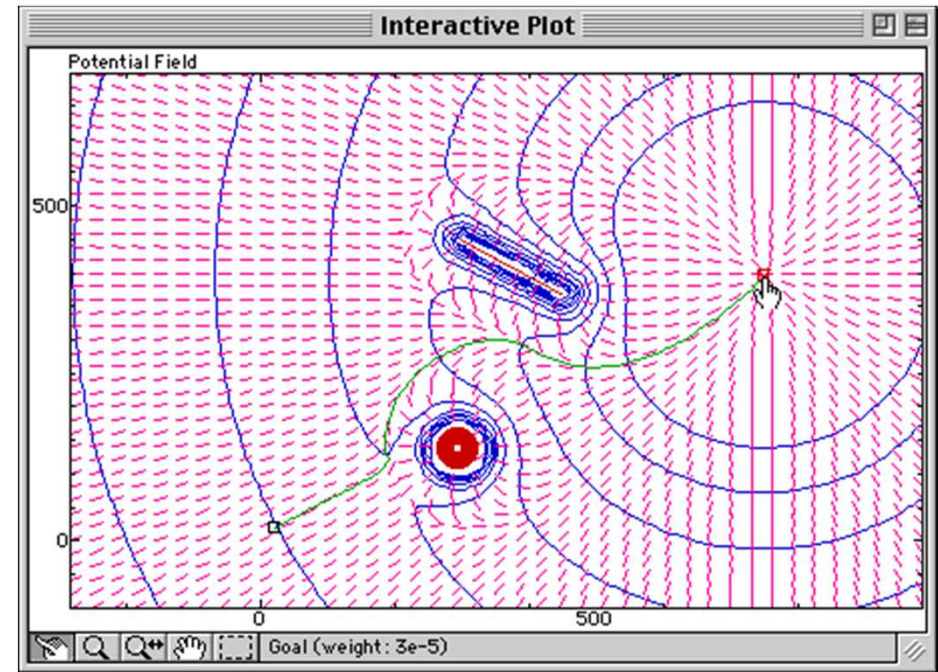
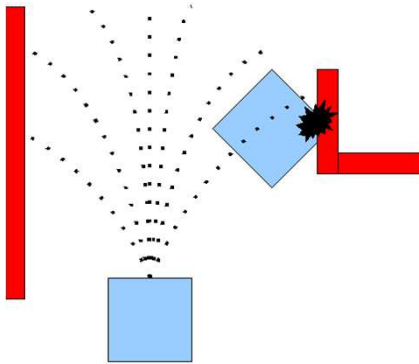


SLAM



# Planificación de movimientos

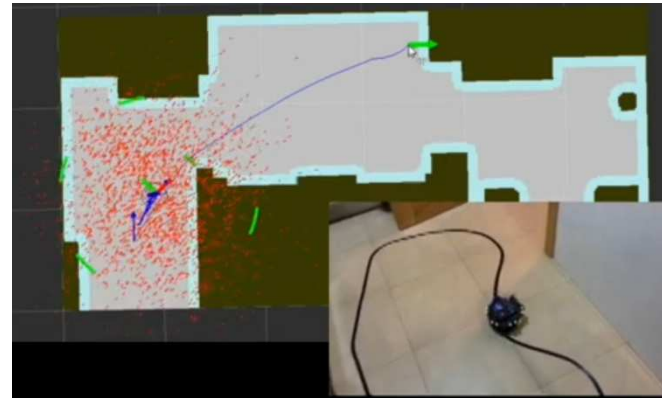
- Planificación local (o planificación reactiva, punto a punto)
  - Enviar el robot desde su posición hasta una ubicación objetivo en un marco local
    - Campos de potencial
      - Campo de fuerzas virtuales
      - Histograma de vectores de campo
    - Dynamic window approach
    - Trajectory Rollout



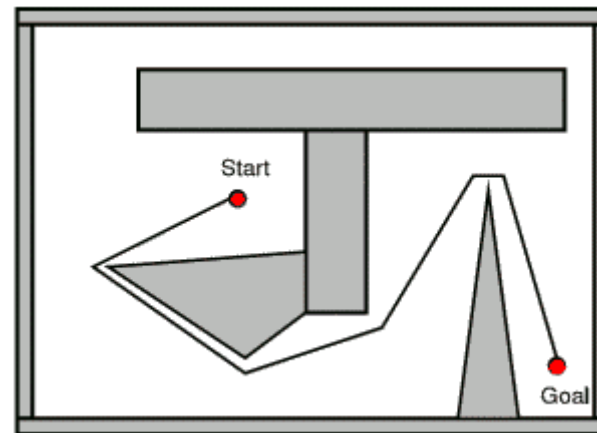
[Navegación local mediante campos de potencial](#)

# Planificación de movimientos

- Planificación de rutas (path planning)
  - hacer que el robot siga una ruta especificada como un conjunto de puntos
    - Búsqueda en grafos
    - A\* y sus variantes

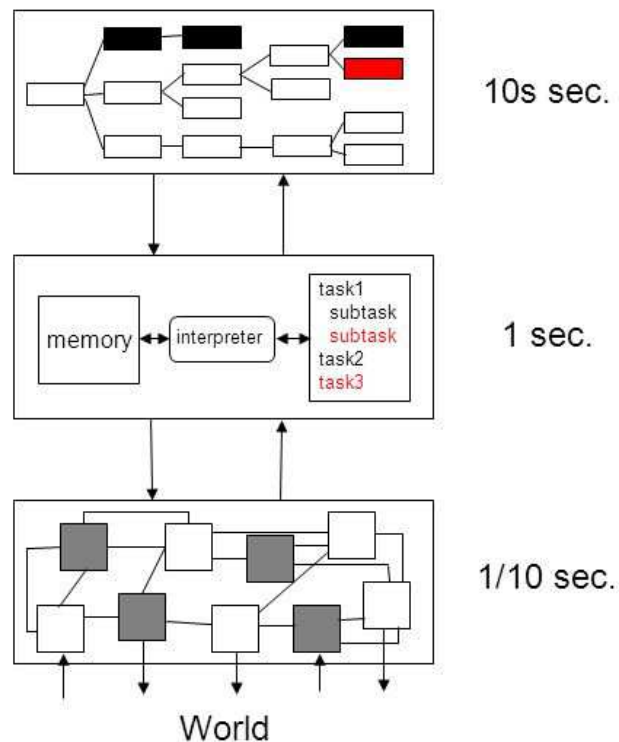


Navegación global (calculando ruta previa)



# Arquitecturas Software

- Arquitectura: Lenguajes, herramientas para escribir programas y organizarlos juntos para controlar un robot.
- Arquitecturas híbridas: combinan técnicas reactivas y deliberativas.



- Arquitectura de 3 niveles:
  - **Nivel Deliberativo:** genera soluciones globales a tareas complejas mediante planificación (de movimientos o the tareas más abstractas). Ciclo de decisión de segundos a minutos.
  - **Nivel Ejecutivo:** establece la unión entre el nivel deliberativo y el reactivo. Acepta directivas del deliberativo. Cico de decisión en 1/10 sgs.
  - **Nivel Reactivo:** control a bajo nivel del robot. Bucle sensor-acción muy acoplado. Ciclo decisión en ms.