

Control y robots manipuladores



Escuela de
Ingeniería
Informática
Universidad de Oviedo



Universidad de Oviedo
Universidá d'Uviéu
University of Oviedo

Cristian González García
gonzalezcristian@uniovi.es

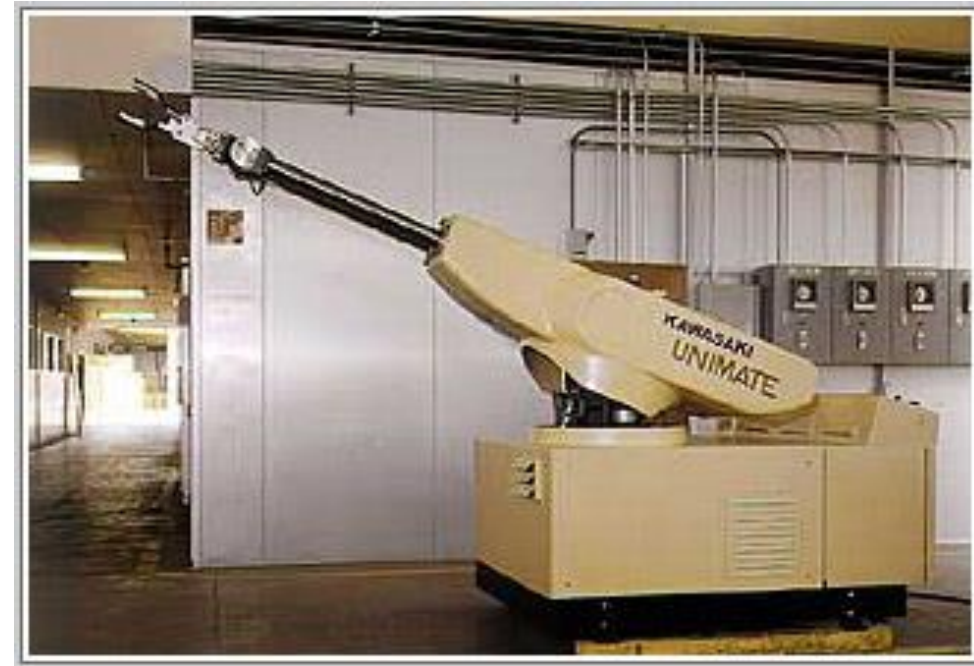
Basado en el material original de Jordán Pascual
Espada

v 1.3.2 Octubre 2022

Historia

Historia I

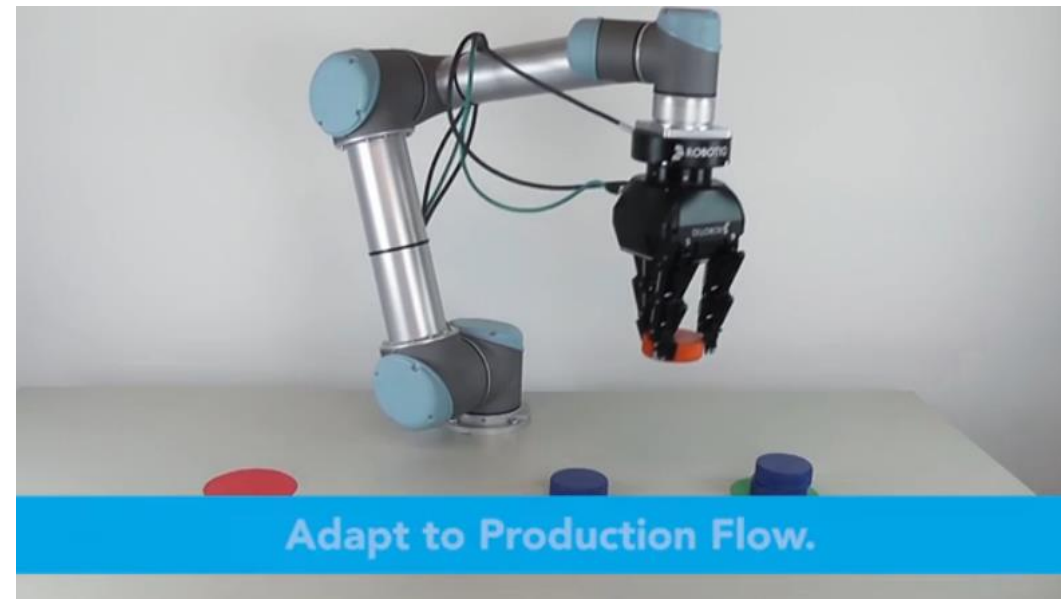
- Los primeros robots realizaban ejecuciones principalmente **secuenciales**
 - Movían sus actuadores en frecuencias prefijadas
 - Motor 1 girar 30°
 - Motor 2 girar 45°
 - Esperar 1000 ms
 - Motor 3 girar 90°
 - Etc.
 - Había que decirles **TODO**



1969 Kawasaki-Unimate 2000
Japón

Historia II

- **Ocasionalmente fueron integrando sensores**
 - Permiten **condicionar acciones** de la secuencia
 - Motor 3 girar 1° -> while (sensor presión < 0)
- **Seguridad**
 - Si colisiona -> STOP
- Un poco de **inteligencia**



Historia III

- Posteriormente **surgieron** muchos **otros mecanismos y sistemas de control**
 - **Representación del entorno**
 - Entender que había alrededor de ellos
 - Con mayor número de sensores
 - Mecanismos de **razonamiento y algoritmos inteligentes**
 - Planear la **consecución de objetivos**
 - Apilar objetos
 - Amazon
 - Que **acciones** son **más adecuadas** a realizar
 - **Aprendizaje automático**
 - IA, Inteligencia de enjambre (Swarm Intelligence)
 - Seguimiento de stock, detección de botellas, movimiento por una zona, comunicación con robots, etc.
 - Etc.
 - Los robots industriales **siguen teniendo un control «simple»**
 - Al menos **la mayoría** de ellos, como los brazos robóticos
 - **Repiten acciones** en forma de bucle
 - Pueden **detectar algunos problemas** con sensores
 - Ej.: ¡La pieza se cayó al suelo!, ¡No hay pieza!, forma más rápida de hacerlo, control de calidad, etc.

Introducción al control de robots

Definición / programación del comportamiento

- ¿Qué tiene que hacer el robot? Depende...
 - Control remoto**
 - Tarea repetitiva** a través de una secuencia fija
 - Tarea que requiere algo de **inteligencia** creando secuencias dinámicas
- ¿Como lo hace?
 - Uso de DSL o lenguajes de alto (POO) / bajo nivel (ASM)



Lenguaje de alto nivel I

- Controla a bajo nivel los elementos del robot
- Requiere la implementación de casi todos los comportamientos
 - Algunos lenguajes son ya conocidos/genéricos (Arduino, Java, etc.)

```

/* Servo pulses */
float bas_servopulse = 1500.0 - (( degrees( bas_angle_r )) * 11.11 );
float shl_servopulse = 1500.0 + (( shl_angle_d - 90.0 ) * 6.6 );
float elb_servopulse = 1500.0 - (( elb_angle_d - 90.0 ) * 6.6 );
float wri_servopulse = 1500 + ( wri_angle_d * 11.1 );

/* Set servos */
servos.setposition( BAS_SERVO, f1( bas_servopulse ));
servos.setposition( WRI_SERVO, f1( wri_servopulse ));
servos.setposition( SHL_SERVO, f1( shl_servopulse ));
servos.setposition( ELB_SERVO, f1( elb_servopulse ));

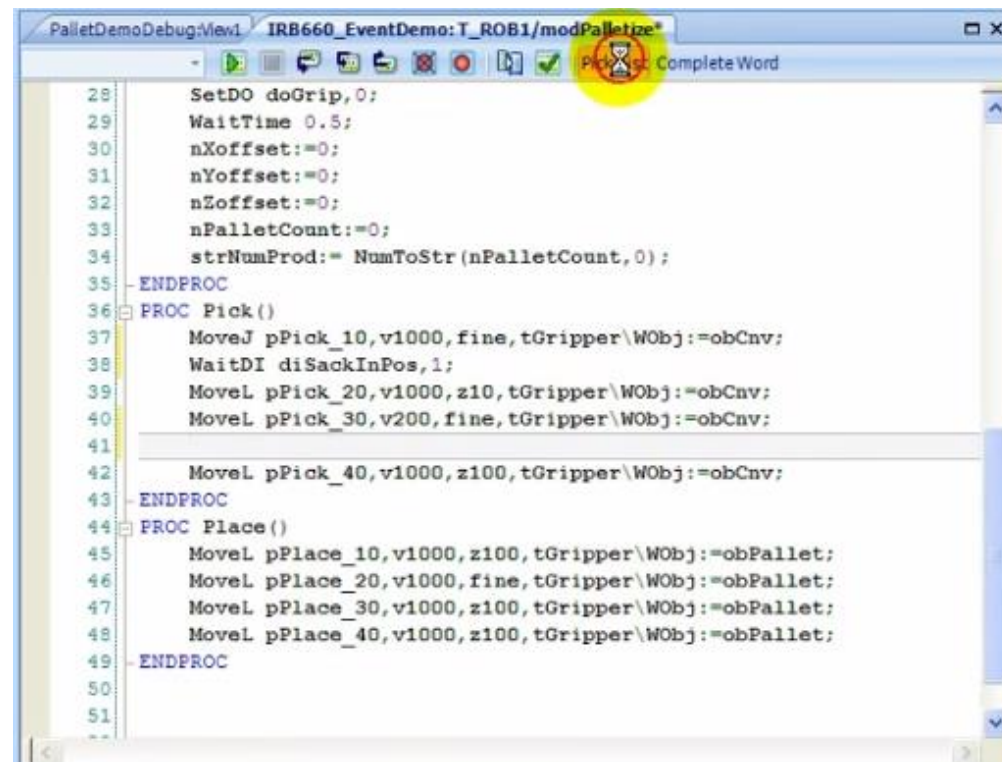
}

/* move servos to parking position */
void servo_park()
{
  servos.setposition( BAS_SERVO, 1715 );
  servos.setposition( SHL_SERVO, 2100 );
  servos.setposition( ELB_SERVO, 2100 );
  servos.setposition( WRI_SERVO, 1800 );
  servos.setposition( WRO_SERVO, 600 );
  servos.setposition( GRI_SERVO, 900 );
  return;
}

```


Lenguaje de alto nivel II

- Muchos fabricantes **desarrollan sus** propios **lenguajes**
 - Pleo, Lego Mindstorm, ABB Rapid, ...



```
PalletDemoDebug:New1 IRB660_EventDemo:T_ROB1/modPalletize*
- [Pick] Complete Word
28 SetDO doGrip,0;
29 WaitTime 0.5;
30 nXoffset:=0;
31 nYoffset:=0;
32 nZoffset:=0;
33 nPalletCount:=0;
34 strNumProd:= NumToStr (nPalletCount,0);
35 -ENDPROC
36 PROC Pick()
37 MoveJ pPick_10,v1000,fine,tGripper\WObj:=obCnv;
38 WaitDI diSackInPos,1;
39 MoveL pPick_20,v1000,z10,tGripper\WObj:=obCnv;
40 MoveL pPick_30,v200,fine,tGripper\WObj:=obCnv;
41
42 MoveL pPick_40,v1000,z100,tGripper\WObj:=obCnv;
43 -ENDPROC
44 PROC Place()
45 MoveL pPlace_10,v1000,z100,tGripper\WObj:=obPallet;
46 MoveL pPlace_20,v1000,fine,tGripper\WObj:=obPallet;
47 MoveL pPlace_30,v1000,z100,tGripper\WObj:=obPallet;
48 MoveL pPlace_40,v1000,z100,tGripper\WObj:=obPallet;
49 -ENDPROC
50
51
```

Lenguaje de alto nivel III

- RAPID creado en 1994 – (Robots de ABB Group)
 - <https://www.youtube.com/watch?v=b5eQaAl0txY>
- Se basa en ARLA
 - <http://hopl.info/showlanguage.prx?exp=7482>
 - https://lilia.dpss.psy.unipd.it/~antonio.calcagni/thc/programming_stats_19.pdf
- Permite
 - Procedimientos
 - Condiciones, bucles...
 - Rutinas con interrupciones
 - Manejo de errores
 - Multitarea
 - Otros.
- Guía: <http://futurecnc.code.arc.cmu.edu/wp/wp-content/uploads/2011/12/RAPID-Reference-Manual-Instructions.pdf>

RAPID, 2020

```
! Example of RAPID code
MODULE Module1
  ! Variable declaration
  VAR num sum:=0;
  VAR num i:=0;
  PROC main()
    WHILE sum<=100 DO
      i:=i+1;
      sum:=sum+i;
    ENDWHILE
  ENDPROC
ENDMODULE
```

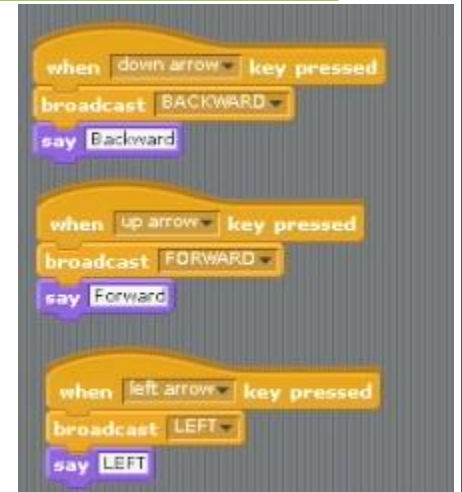
ARLA, 2019

Example 3: Swap values of two variables

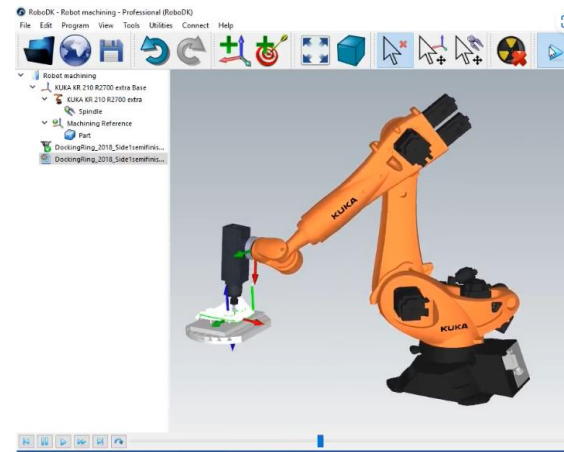
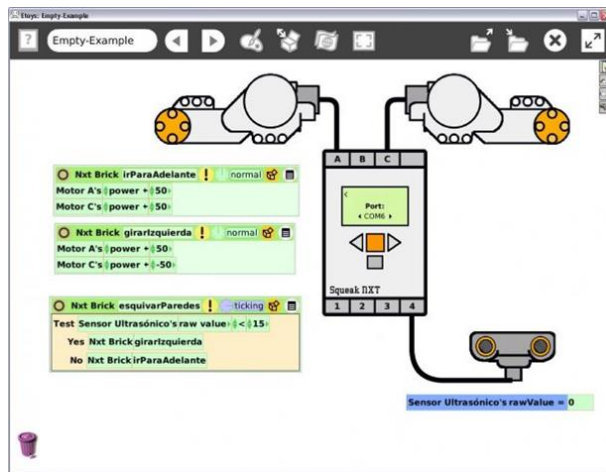
```
a := NUMERIC
b := NUMERIC
c := NUMERIC
BEGIN
  a = read("Write a number")
  b = read("Write another number")
  c = a
  a = b
  b = c
  print(a,b)
END
```

Lenguaje de dominio específico (DSL)

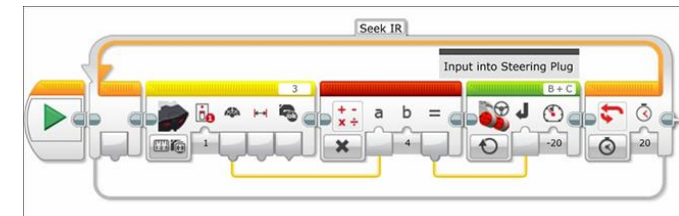
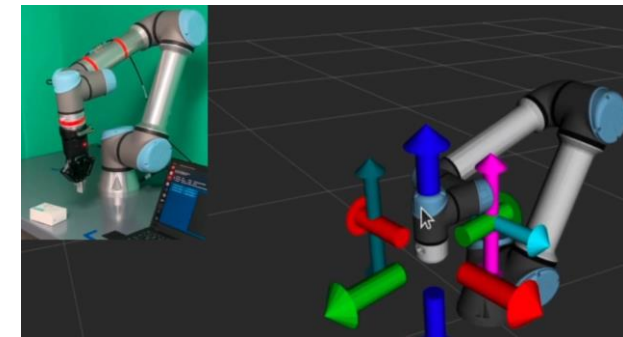
- Especifica **comandos o instrucciones de alto nivel**
 - Creados por otros y reaprovechados: **Scratch**, **Midgar**, **Bitbloq**
- Pueden ser **textuales o gráficos (DSL)**
 - **Pensados** en muchos casos **para ofrecer abstracción a operarios**
 - Creados por el propio fabricante: **Lego Mindstorm**, **Kuka**, **ROS** (<https://www.ros.org/>), etc.



Scratch:
https://www.youtube.com/watch?v=w0_I5PCxf8w



Kuka: <https://robodk.com/blog/kuka-programming-guide/>



Lego Mindstorm NXT: https://es.wikipedia.org/wiki/Lego_MindStorm_NXT#/media/Archivo:Auto_PE.png

<http://blog.electricbricks.com/2018/11/lenguajes-de-programacion-de-mindstorms-ev3/>

Tipos de control robótico

Tipos de control robótico

- Operarios humanos
 - **Teleoperación**
 - Da Vinci, Mars Rovers Curiosity
- Secuencia fija de movimientos
 - **Playback**

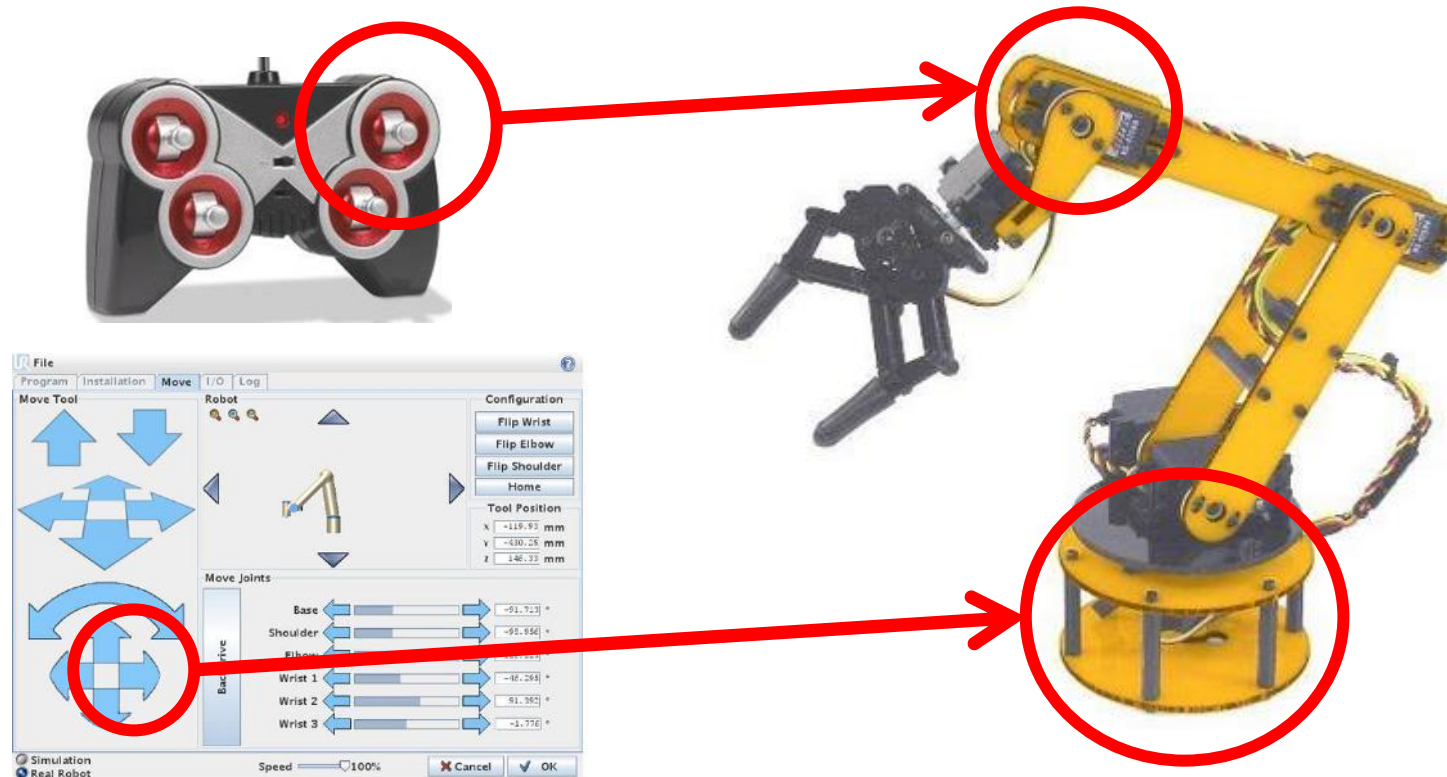
Teleoperación

Teleoperación

- El **robot** es **directamente controlado** por un operario
 - Mando a distancia, cabina, panel de control, etc.
- Controla la posición, la velocidad, la fuerza ejercida, etc.
 - **Depende de cada robot**
 - No todos ofrecen el mismo control ni las mismas opciones
- Diferentes **arquitecturas de control**
 - En función del intercambio de información entre el operario y el robot
 - El operario se denomina «**maestro**» y el robot «**esclavo**»
 - **Control** por: fuerza, velocidad, posición, etc.

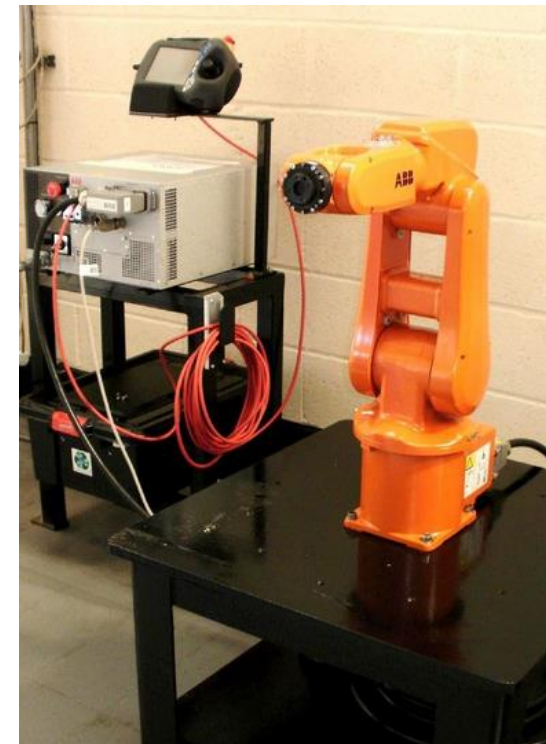
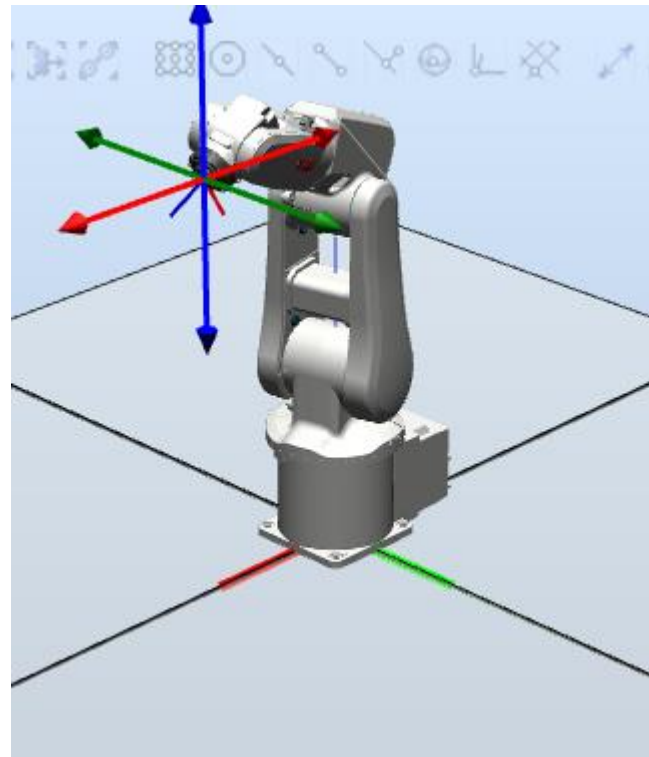
Arquitecturas I – Fuerza/Velocidad

- **Especifica la fuerza / velocidad** que se transmite a los actuadores



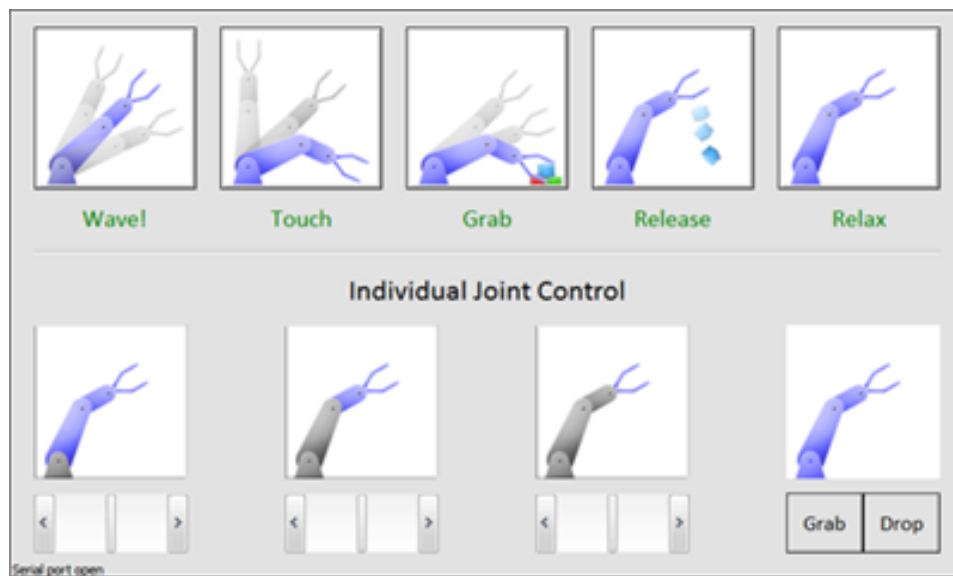
Arquitecturas II – Posición

- **Especifica** los puntos a **donde debe moverse**
 - Cada posición puede mover varios actuadores (ej.: 3 motores)



Arquitecturas III – Comandos

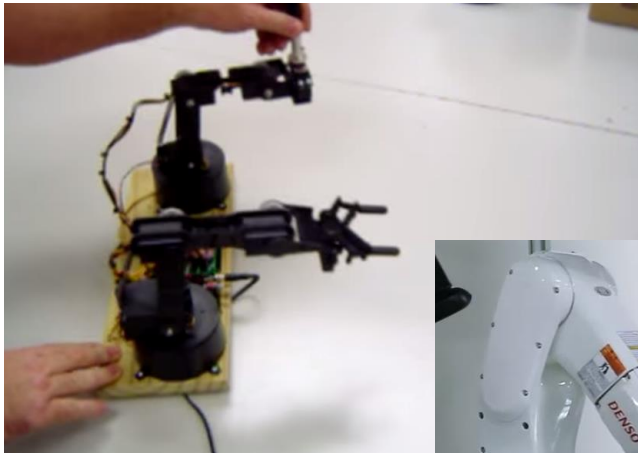
- **Especifica comandos**
 - Instrucciones **de alto nivel**
 - Cada instrucción puede requerir varios movimientos



Arquitecturas IV – Maestro-Esclavo

- **Especifica la posición** de los actuadores en un «esclavo»

<https://youtu.be/zZwnyZPkyIk?t=7>



<https://youtu.be/KrQsczjMeTw>



<https://youtu.be/5FmQ9tiXe48>

Otros ejemplos de teleoperación I

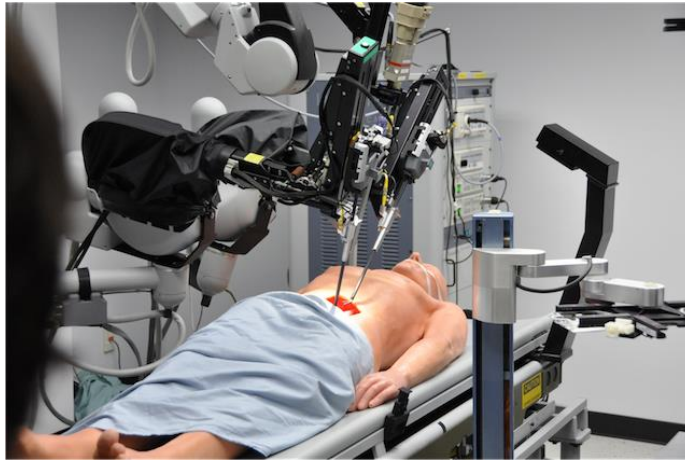
- ◉ Quirúrgicos
- ◉ Industriales (Pocos)
 - ◉ Cucharas de fundición
 - ◉ <https://youtu.be/3UL3kBQx4eo>
- ◉ Manipular productos químicos (Fukushima y Chernóbil)
- ◉ Militares (Drones, etc.)
- ◉ Asistencia personal
- ◉ Otros
 - ◉ Flex Pendant – Robot Studio - ABB
 - ◉ <https://youtu.be/RSjCAmaS8ac>



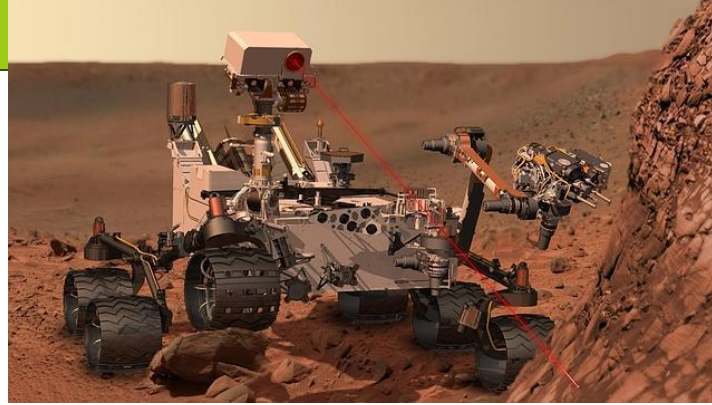
Flex Pendant



Otros ejemplos de teleoperación II



Problemas



- Mars Rover Curiosity -> **Retrasos en la señal** maestro-esclavo
 - **Retraso en las comunicaciones** entre la Tierra y Marte
 - **Lleva una IA que elige** que perforar y así realizar más experimentos
 - Se integró mediante una actualización software que se descargó
- **Se está desgastando**
 - Ruedas
 - Tuvo que «autocambiarse» el ordenador de abordo por problemas
- Se hace selfies
 - Comprobar estado del robot
 - <https://youtu.be/4tqLk-Wl0Fo?t=7>
- Datos
 - <https://twitter.com/MarsCuriosity>
 - 9 años en Marte (6-08-22)
 - Previsión de 23 meses de vida útil
 - Velocidad: 130m/h = 36,11 km/h
 - Coste: 2.600 millones\$
 - Plutonio -> transformando calor en electricidad (14 años)
 - 29 km recorridos (6-08-22)
 - Analizó 41 rocas (6-08-22)
- 2021: Mars Rover Perseverance



Teleoperación con Arduino

Teleoperación con el Arduino

- **Mecanismos de interfaz** de usuario / **sensores** conectados a la placa
 - Sistema para: controlar el robot, grabar tareas, reproducirlas, etc.



Puerto COM

- Conexión al PC, por **puerto COM**
 - La mayor parte de GPLs pueden acceder al puerto COM
 - Desarrollar una aplicación software
 - <https://playground.arduino.cc/Interfacing/Java>

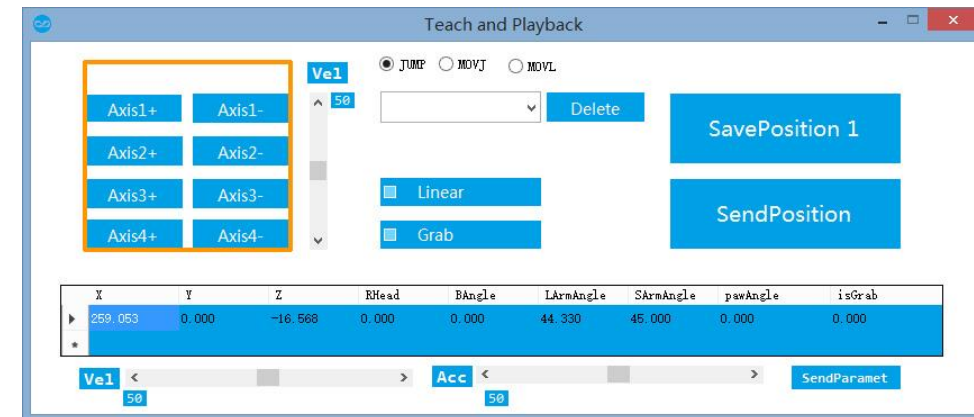
```

public void serialEvent(SerialPortEvent arg0) {
    if (arg0.getEventType() == SerialPortEvent.DATA_AVAILABLE) {
        try {
            byte byteLeido = (byte) input.read();
            lectura += new String(new byte[] { byteLeido });

            if ( new String(new byte[] { byteLeido }).equals("\n") ) {
                System.out.println("Arduino dice: " + lectura);
                lectura = "";
            }
        } catch (Exception e) {
            System.out.println("Error F " + e.getMessage());
        }
    }
}

```

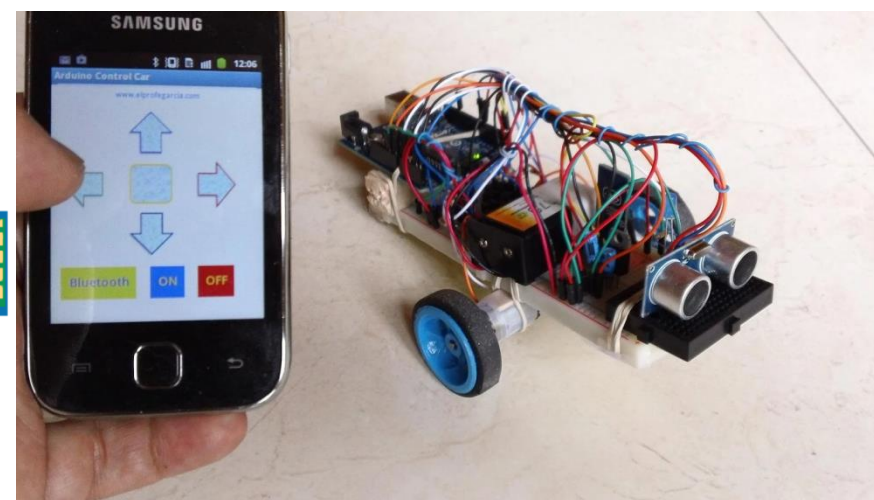
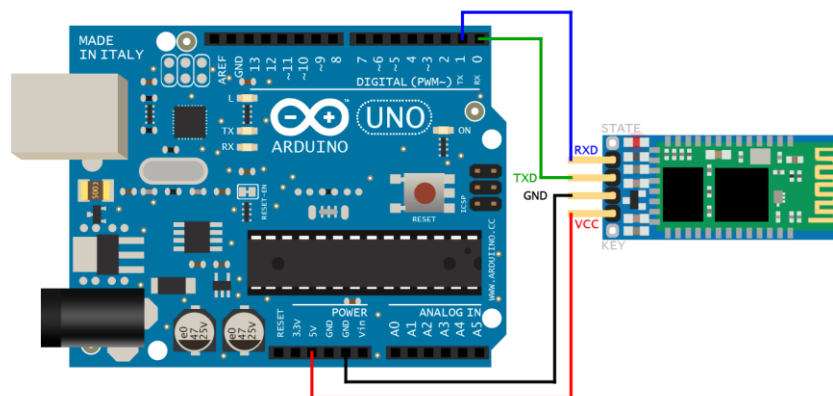
Fragmento Java



Interfaz gráfica

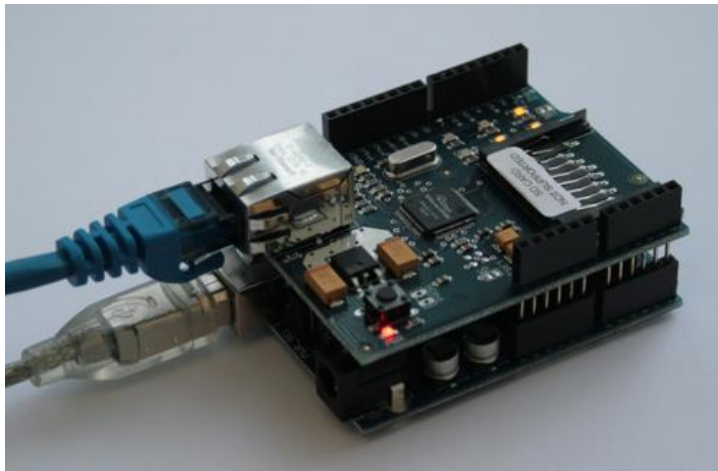
Bluetooth

- Comunicación inalámbrica de rango cercano



Lan / Internet

- Crear un **Servidor Web en la placa**
 - Se le envían instrucciones a través de internet
 - Servicios Web / Interfaz Web



Control robótico – Playback

Playback

- El robot maneja una **secuencia fija de movimientos que repite**
- **Común en robots manipuladores industriales**
 - Ej: soldadura, pintura, etc.
- **Se le especifica la secuencia** de movimientos (opciones)
 - Controlando el robot en «**modo grabación**»
 - **Moviendo las partes del robot** en modo grabación
 - **Con «lenguajes de programación»** de alto o bajo nivel
 - Comandos / instrucciones

Modo grabación

- **El operario realiza una tarea moviendo o controlando el robot**
- **Mientras realiza el control -> se registra toda lo necesario para la actividad**
 - Ej.: velocidades, posiciones de los actuadores, tiempos, etc.
 - Igual la velocidad siempre es fija...
 - Datos necesarios para poder repetir el proceso
 - Grabar a velocidad normal o más lento
- **Modificación o configuración** de la tarea grabada
 - Modificar cualquier acción para no tener que repetir la grabación o usarlo como método de afinación
 - Ej.: aumentar / reducir la velocidad
 - Cambiar puntos de la secuencia, etc.

Ejemplo I

- <https://youtu.be/blcOHQ97Eg0>



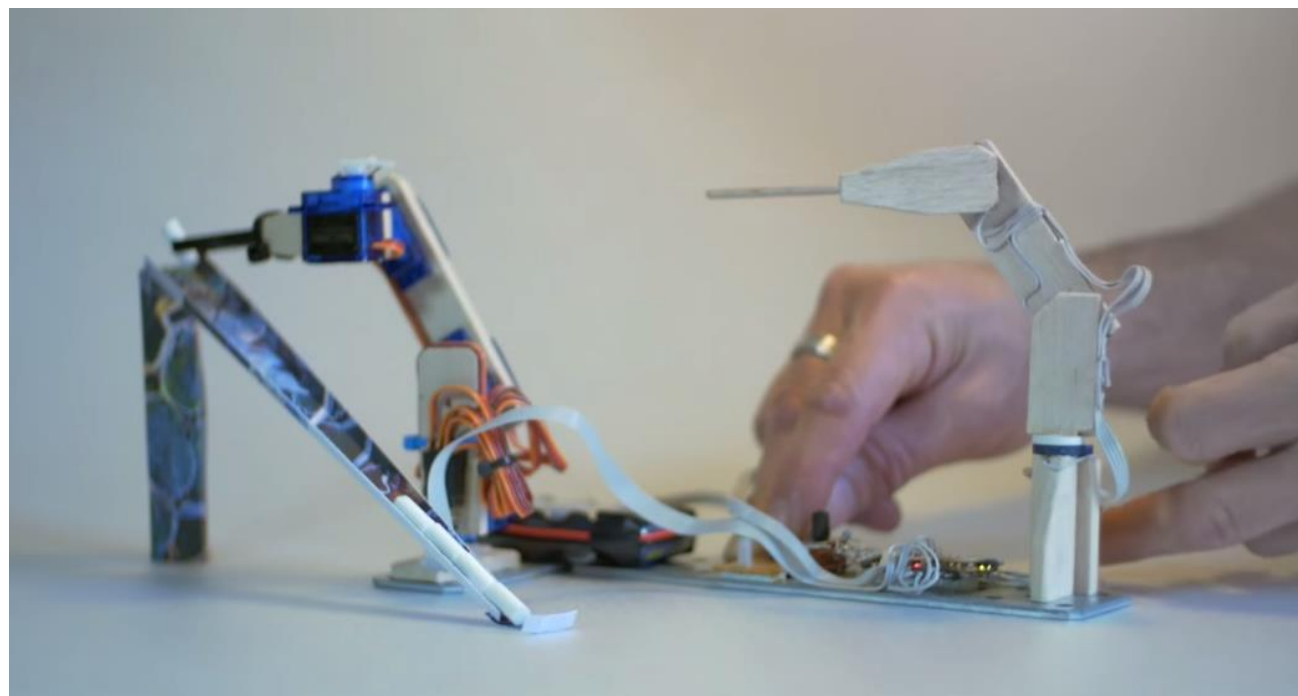
Ejemplo II

- Baxter
 - Ordenador externo con ROS: <https://youtu.be/Y3uAe0fKEVM>
 - Todo con el robot: <https://youtu.be/weBPQOT4Ymg>



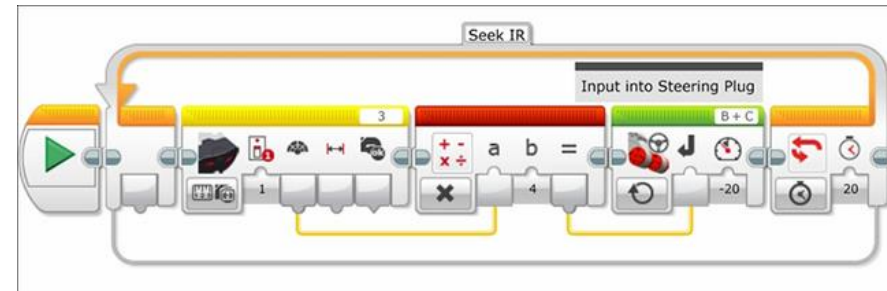
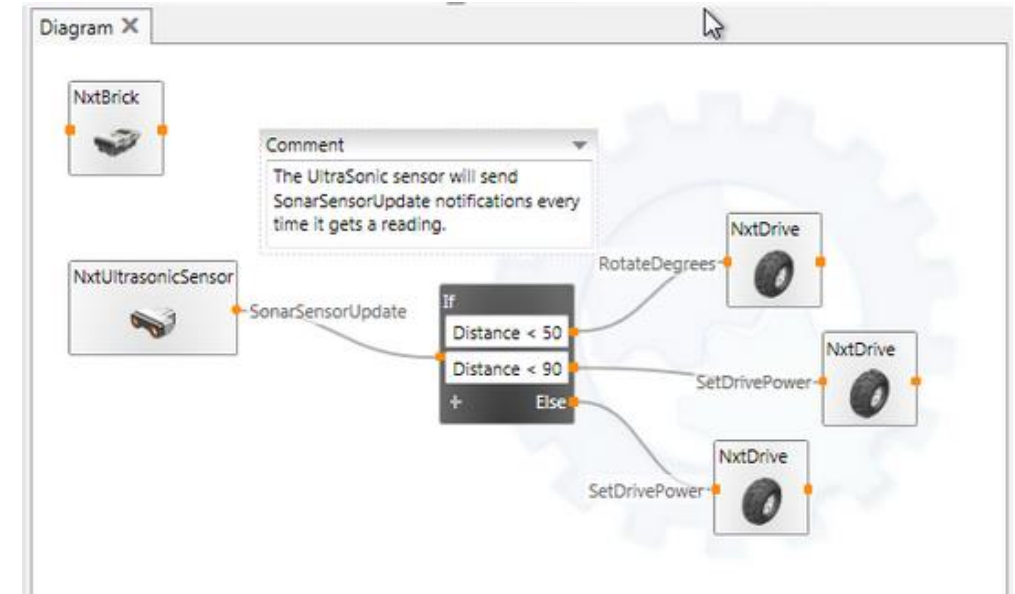
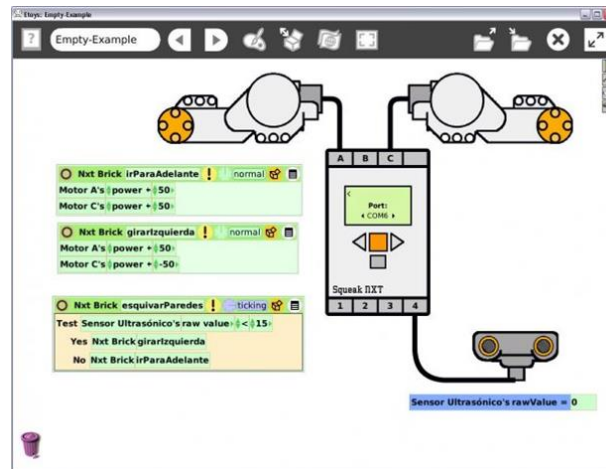
Ejemplo III – Maestro - Esclavo

- <https://youtu.be/bLnAJ-mSEIE>
- Código: <http://letsmakerobots.com/robot/project/micro-servo-robot>
- <https://web.archive.org/web/20170110063409/http://letsmakerobots.com/robot/project/micro-servo-robot>



Programación de tareas I

- Comandos / instrucciones
- De forma **gráfica o textual** (MDE y DSL)



<http://blog.electricbricks.com/2018/11/lenguajes-de-programacion-de-mindstorms-ev3/>

Programación de tareas II

- **Sobre PCs o dispositivos especializados**

- Ej: FlexPendant

- <https://youtu.be/RSjCAmaS8ac>



- **Programación de alto o bajo nivel**

- Bajo nivel: **especifican el control de los actuadores**

- Ej: mover motor 1 -> 30 grados.

- Alto nivel: **especifican tareas de alto nivel mas abstractas**

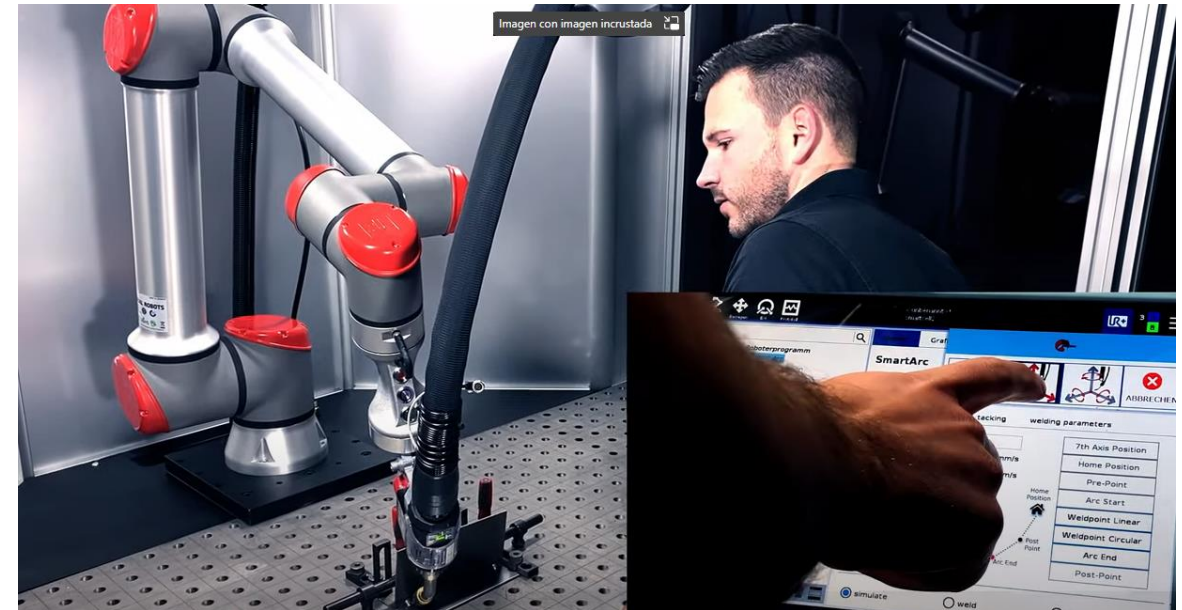
- Ej: recoger la pieza situada en P1 (X:20, Y:22, Z:4)

Ejemplo IV - Soldador

<https://youtu.be/6BtgdmH8hrk?t=21>



https://youtu.be/_seOSvtrXNY



Ejemplo V - Cocina

- Automatización de tareas
 - Repetitivas, costosas o peligrosas para los humanos, comodidad, etc.
 - <https://youtu.be/5F63q4TB5YM>



Preguntas tema 5

○ <https://forms.office.com/r/P2MTpgFWCc>



Puntos de vista de control del robot

Puntos de vista de control del robot

- **Puntos de vista de control** del robot
 - Control reactivo
 - Control de deliberación
 - Control híbrido
 - Control basado en el comportamiento

Robot manipulador vs móvil I

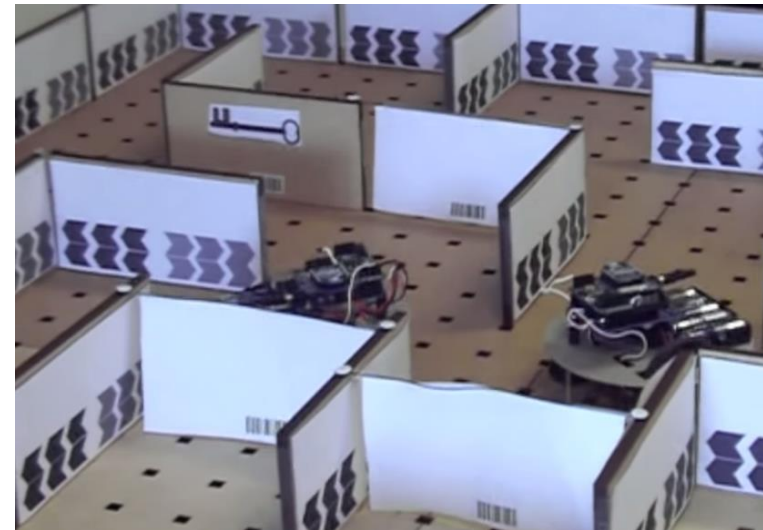
- Las secuencias variables son **más comunes en los robots móviles que en los manipuladores**
- Manipulador**
 - Ordena** cubos en base a unos criterios (Ej.: color, tamaño, etc.)
 - Detecta** si el objeto que está manipulando tiene un error y lo desecha



Robot manipulador vs móvil II

○ Móvil

- **Buscar** un objeto
- **Encontrar** la salida
- **Limpiar** una habitación
- **Esquivar** obstáculos
- **Mover/trasladar** objeto



Clasificación de los robots

- Una **clasificación** sobre el control de los robots es la **de Mataric** [1]
- Se basa en
 - **Como los robots consiguen información** del entorno en el que se encuentran (sensores)
 - **Procesan** dicha información
 - **Toman una decisión**
 - **Actúan** al respecto
- Esta clasificación se basa en que **no todos los robots piensan**
 - **Algunos** disponen ya de reacciones **preprogramadas** para ejecutarse
 - **Otros piensan mucho para actuar un poco**
- **La complejidad del ambiente está directamente relacionada con la complejidad del control**
 - **Si el robot necesita actuar rápido en un ambiente muy dinámico**, esto será una tarea muy complicada
 - En medio del océano, en una guerra, sitios con personas, etc.
 - **Si el robot no tiene que responder rápidamente y se encuentra además en un ambiente no tan dinámico y cambiante**, será más fácil
 - En casa y sin mascotas

Clases

- Hay **cuatro clases**
 - Control **reactivo**
 - No pienses, reacciona
 - Control de **deliberación**
 - Piensa y después actúa
 - Control **híbrido**
 - Piensa y actúa independientemente en paralelo
 - Control **basado en el comportamiento**
 - Piensa en la manera de actuar
- **Cada una pensada para un tipo de trabajo concreto**
- Todas tienen sus **pros y sus contras**

Control reactivo I

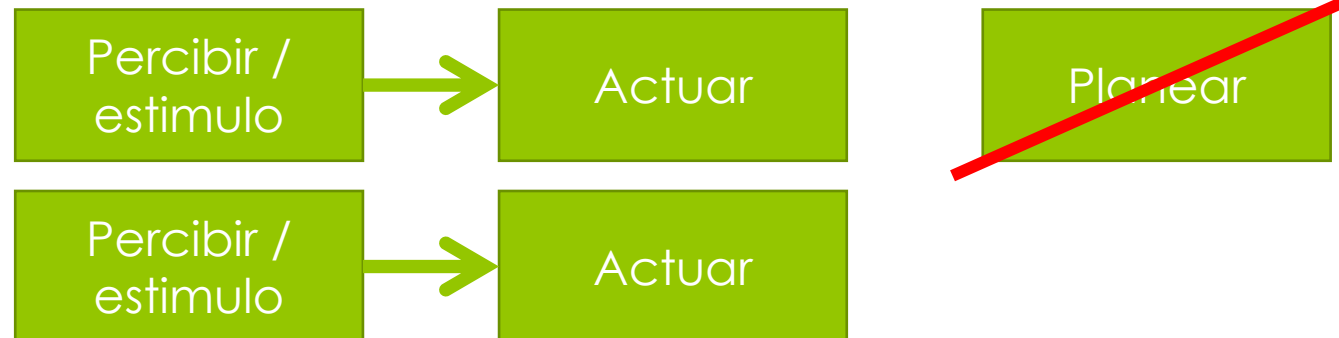
- **No pienses, reacciona**
- Los robots pueden **reaccionar a los cambios de una forma rápida**
 - **Poseen reglas predefinidas** para las diferentes posibles entradas **y se ejecutan instantáneamente**
- **Las entradas** sensoriales **y las salidas** de los actuadores **están fuertemente acopladas**
 - Les permite poder **reaccionar de una manera rápida** [2]
- Este enfoque **requiere de información ambiental** para tener un control efectivo
- **Los robots** que lo utilizan **no suelen mantener guardada mucha información o aprender** en el tiempo
- ✓ ◦ **El control reactivo es bueno para ciertas tareas**
- ✗ ◦ Cuando se necesita que el robot aprenda y tenga memoria este tipo de control es insuficiente
 - Ej.: robot aspiradora/segador

Control reactivo II

- Control **inspirado en los mecanismos de estímulo-respuesta de los animales**

- Ej.: insecto

- No percibe comida -> vuela ¿?
- Percibe comida -> vuela hacia la comida
- Percibe peligro -> escapa



- Correspondencia entre lo que se percibe y como se actúa**

- Para cada tipo de percepción se realiza una acción

Control reactivo III – Ventajas y desventajas

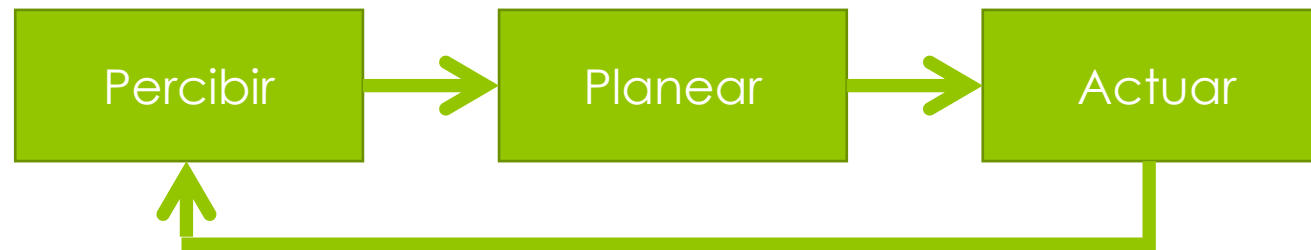
- Algunas **ventajas**
 - **Rápido**: funciona bien en tiempo real
 - **Sencillo** de «modularizar»
 - **Puramente reactivos**: no tienen representaciones internas del entorno
 - «No miran hacia el futuro»
 - Lo básico para «sobrevivir» y realizar su tarea
- Algunas **desventajas**
 - No utilizar un plan **imposibilita realizar algunas tareas**
 - **No hay resolución de conflictos** (varias percepciones simultaneas)
 - Exclusión de percepciones, etc.
 - Aspiradora atascada en una mesa
 - **Requiere muchas pruebas**
 - **Puede ser complejo incluir sistemas de aprendizaje**

Control de deliberación I

- **Piensa y después actúa**
- **Las acciones** son tomadas
 - **A partir del razonamiento de la lectura de todas las entradas** sensoriales
 - **Y del conocimiento almacenado internamente del mundo externo**, sobre la que se basa el cálculo del resultado de salida
- **Permite al robot planificar los pasos de su decisión y saber cómo podría afectarle esto en el futuro** mediante la **predicción de posibles resultados**
- ✓ ○ Este proceso **permite al robot actuar estratégicamente** seleccionando la **mejor decisión** para cada situación
- ✗ ○ **Muy pocos robots son puramente deliberativos**
 - Esta planificación **necesita de una gran computación interna** para lograr la IA necesaria para computar todo el proceso
 - Si se encuentra **en un ambiente muy dinámico y «ruidoso» puede ser imposible tomar una decisión**
 - Salida de un parking con un paso de peatones
 - Ej.: robot móvil, resolución de laberintos, algunas aspiradoras, etc.

Control de deliberación II

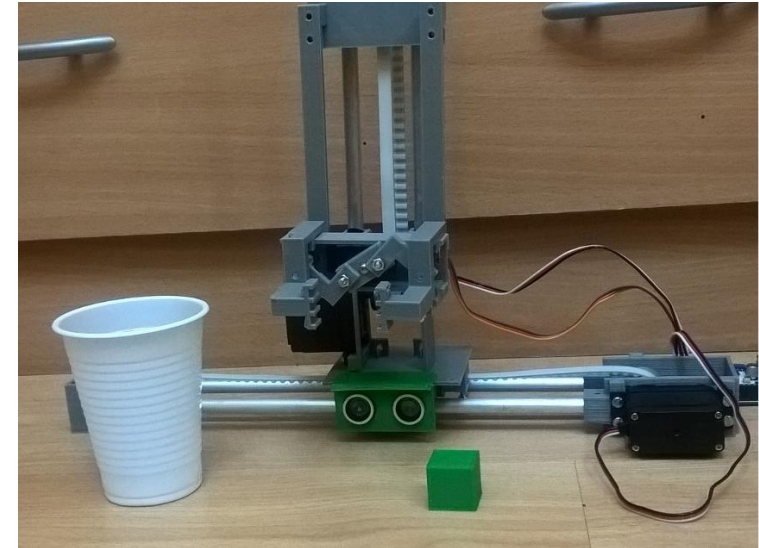
- Control **inspirado en** los mecanismos de **pensamiento de las personas**



- Percibir**
 - Obtiene la información necesaria del entorno / interna, mediante sensores
- Planear**
 - Partiendo de la información, del estado actual y objetivos
 - Realiza un plan
 - Secuencia de acciones para conseguir un objetivo
 - La planificación **requiere computación y algoritmos** (IA)
- Actuar**
 - Da instrucciones a los actuadores para llevar a cabo las acciones
 - Las acciones están **orientadas a cumplir un objetivo** (plan)
 - Elegir un camino esquivando un objeto conociendo previamente el plano del lugar

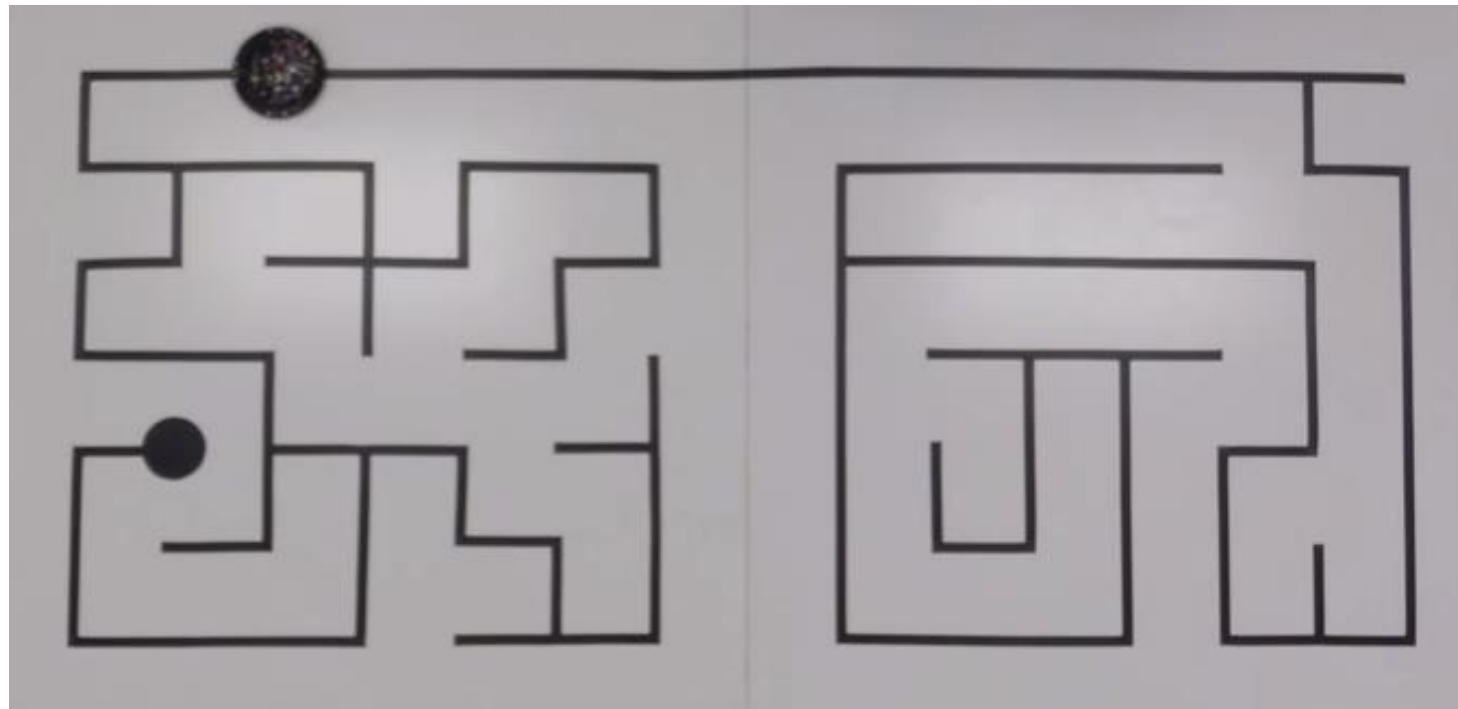
Control de deliberación III – Ejemplo I

- Guardar el cubo en el vaso
 - **Percepción**
 - Posición del cubo y del vaso
 - **Plan**
 - ¿Dónde están los elementos?
 - ¿Qué «acciones» puedo realizar?
 - ¿Dónde está el objeto objetivo?
 - ¿Qué secuencia de acciones me lleva al objetivo?
 - **Actuar**
 - Enviar a los actuadores las señales necesarias para realizar cada acción
 - **Repetir proceso**
 - Percepción -> Plan -> Actuar



Control de deliberación IV – Ejemplo II

- Analiza todos los posibles caminos
- Calcula el camino óptimo
- Resuelve el laberinto utilizando el camino óptimo



Control de deliberación V – Desventajas

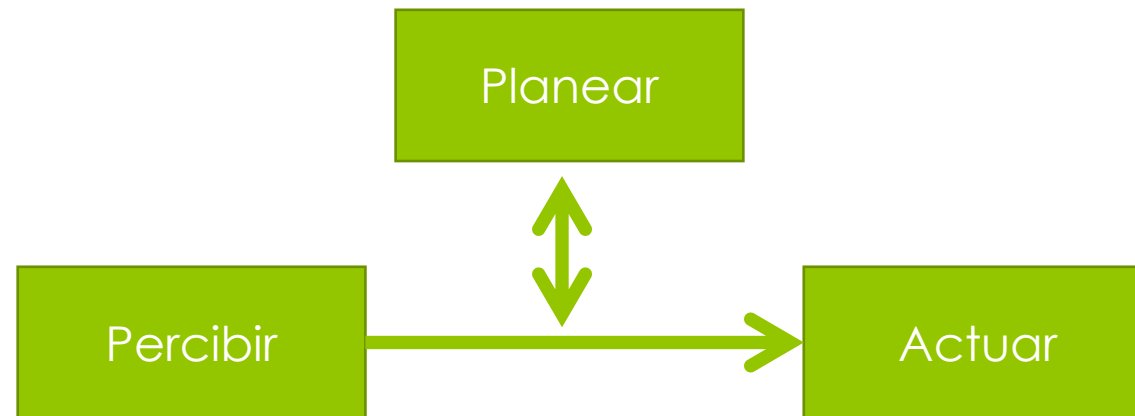
- Los algoritmos de planificación suelen ser «**costosos**»
 - Tienen que **replanificar**...
 - ... si el mundo cambia
 - ... si no se consigue el objetivo
 - ... igual eternamente si el ambiente es muy dinámico
- **No maneja bien la incertidumbre**
 - Todo debe estar considerado en el plan
 - En caso contrario hay que replanificar
 - **Si el entorno no está «controlado» pueden ser poco efectivos**
 - O incluso nunca actuar...
- La **representación** del problema suele ser **muy compleja**
 - Representación del problema -> necesaria para planificar
 - Estados del problema, acciones que puede realizar, efectos de las acciones, el ambiente que le rodea, etc.

Control híbrido I

- **Piensa y actúa independientemente en paralelo**
- ✓ ○ Es la **fusión del control reactivo y del control de deliberación**
 - Coge lo mejor de ambos
- Los **objetivos**
 - A **largo plazo** se rigen por un elemento de **deliberación** para tomar decisiones eficaces
 - Las **acciones inmediatas y en tiempo real** son controladas por un **elemento reactivo**
- La dificultad se encuentra en que se **utiliza el control reactivo para esquivar obstáculos y tomar otras decisiones** que deben de ser **inmediatas**, mientras que **el componente de deliberación toma decisiones más a largo plazo**
- ✗ ○ Es necesario **un sistema que coordine ambas partes** para que tengan un beneficio mutuo
 - **El sistema reactivo anula el de deliberación** cuando se necesita tomar una decisión inmediata
 - **El sistema deliberativo debe informa al reactivo** de poder guiar al robot hacia **trayectorias más eficientes**
 - Este componente intermedio es el más difícil de diseñar en este tipo de controles
- Ej.: obstáculo en medio del camino, coche con calle cortada, aspiradora

Control híbrido II

- Combinan operaciones deliberativas y reactivas
 - **Deliberativas**: planificación a alto nivel
 - **Reactivas**: entran en funcionamiento ante determinadas percepciones



Control híbrido III – Módulos

- **Los módulos** deliberativos **funcionan de forma relativamente independiente** a los reactivos
- **La** parte de la **planificación requiere conocimiento del mundo**
- **Módulos reactivos**
 - Se basa en la situación **presente**
- **Módulos deliberativos**
 - Se basan en el **modelo del mundo**
 - **Razona** sobre **pasado, futuro** y el objetivo
 - Ej.: bache en el camino puede significar más baches, y ya sabes como esquivarlo

Control híbrido IV – Ejemplo

- **Requiere una organización de responsabilidades**

- ¿Qué acciones tienen prioridad?
- ¿Cómo alteran el plan los módulos reactivos?

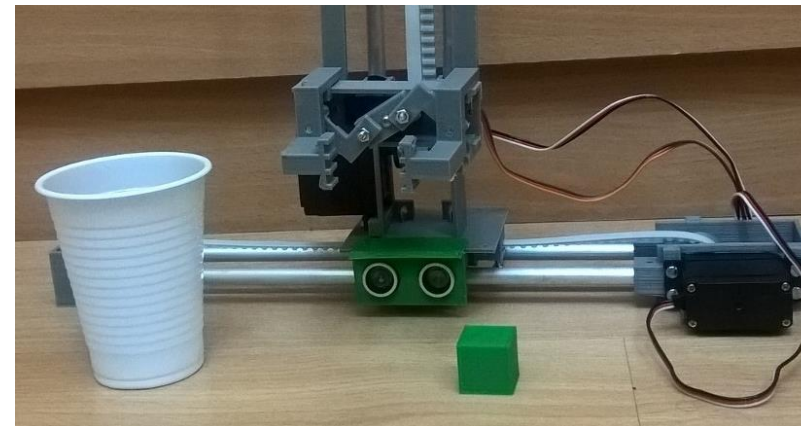
- **Ejemplo: guarda cubo en el vaso + detecta si se caen por el camino**

- **Deliberativo**

- Percepción
- Construir plan

- **Reactivo:**

- **Cubo se cae** -> (1) recoger
- (2) Rehacer plan



Control basado en el comportamiento I

- **Piensa en la manera de actuar**
- **Basado en la biología**
 - Se basa en el comportamiento de los animales con sus entornos
- **Se guardan los patrones**, conocidos como comportamientos, **de las actividades que surgen cuando el robot interactúa con el entorno**
- **Construcción**
 - **Empieza con los comportamientos más simples**
 - Se comienza con comportamientos de supervivencia, como evitar colisiones
 - Esto es un comportamiento que **va mapeado directamente a unas determinadas entradas sensoriales**
 - **Sigue hacia los comportamientos más complejos**
 - **Se mejoran los comportamientos**
 - **Se añaden otros mucho más complejos**, como seguir una pared, perseguir un objeto en movimiento, explorar o buscar el retorno a casa, ...
- ✓ ○ **Puede almacenar las representaciones**, que hace que sea mucho más poderoso
- ✓ ○ **Utiliza un sistema distribuido para almacenar los comportamientos**
 - Permite que, si un robot necesita crear una planificación de futuro, el mismo robot pueda **acceder al sistema distribuido para obtener una mejor información** acerca de cómo actuar
- ✗ ○ **Son los más complicados** de desarrollar

Varios - Publicidad

- Videos publicitarios de robots Kuka:
 - <https://youtu.be/tIIJME8-au8>
 - <https://youtu.be/lv6op2HHluM>
 - Making of: <https://youtu.be/c2NeW9o5G6s>

Referencias

Referencias

- Casi todo el contenido de los comportamientos pertenece al capítulo 12 de:
 - González García, C., 2017. MIDGAR: Interoperabilidad de objetos en el marco de Internet de las Cosas mediante el uso de Ingeniería Dirigida por Modelos. University of Oviedo. doi:10.13140/RG.2.2.26332.59529
 - https://www.researchgate.net/publication/314188769_MIDGAR_interoperabilidad_de_objetos_en_el_marco_de_Internet_de_las_Cosas_mediante_el_uso_de_Ingenieria_Dirigida_por_Modelos
- 1. M. J. Matarić, "Situated robotics," *Encycl. Cogn. Sci.*, pp. 25–30, 2002.
- 2. R. A. Brooks, "A robust layered control system for a mobile robot," *IEEE J. Robot. Autom.*, vol. 2, no. 1, pp. 14–23, Jun. 1986.

Control y robots manipuladores



Escuela de
Ingeniería
Informática
Universidad de Oviedo



Universidad de Oviedo
Universidá d'Uviéu
University of Oviedo

Cristian González García
gonzalezcristian@uniovi.es

Basado en el material original de Jordán Pascual
Espada

v 1.3.2 Octubre 2022