6 ROBOTS MANIPULADORES - ACTIVIDADES

Software para Robots
Jordán Pascual : pascualjordan@uniovi.es

Introducción

Esta actividad corresponde con el conjunto de **actividades finales del bloque 1** que se desarrollará durante las sesiones de prácticas 5 y 6. Se desarrollará en los grupos de 3-5 que ya han sido formados anteriormente.

- Robots manipuladores Parte 1 (actividades 1 - 5)
- Robots manipuladores Parte 2 (actividades 6 -10)

Durante las dos prácticas se presentará un grupo actividades relativas a robots manipuladores (se deben tratar como un único conjunto, aunque se presentará la mitad en cada una de las sesiones). En total las dos partes contendrán 10 actividades, de las cuales el grupo debe elegir las que quiere realizar en función de sus intereses (Puede elegir realizar actividades de cualquier parte indistintamente, ejemplo un grupo podría hacer las actividades 1,2,3,4, otro las 6,7,8,9, etc.).

Las actividades tienen un peso máximo sobre la nota del bloque 1 de 6,5 puntos, cada actividad tiene una puntuación asociada en función de su complejidad.

♣ B.1.1 – Programación en Arduino (Parejas) – 0,75 puntos
♣ B.1.2 – Control de sensores y actuadores (Parejas) – 0,75 puntos
♣ B.1.3 – Diseño de modelos 3D (Grupo) – 1,5 puntos
♣ B.1.4 – Actuadores lineales (Parejas) – 1,5 puntos
♣ B.1.5 (practicas 5-6) – Robots manipuladores (Grupo) – 6,5 puntos

Total: 11 Puntos.
El bloque 1 tiene un peso de 40% en el total de la asignatura.

Se deben seleccionar los apartados que se desean realizar, en caso de que la nota de los apartados sobrepase los 6,5 puntos será truncada.

5.6 Detección y recogida simple de cubo con manipulador cartesiano (1,6 Puntos)

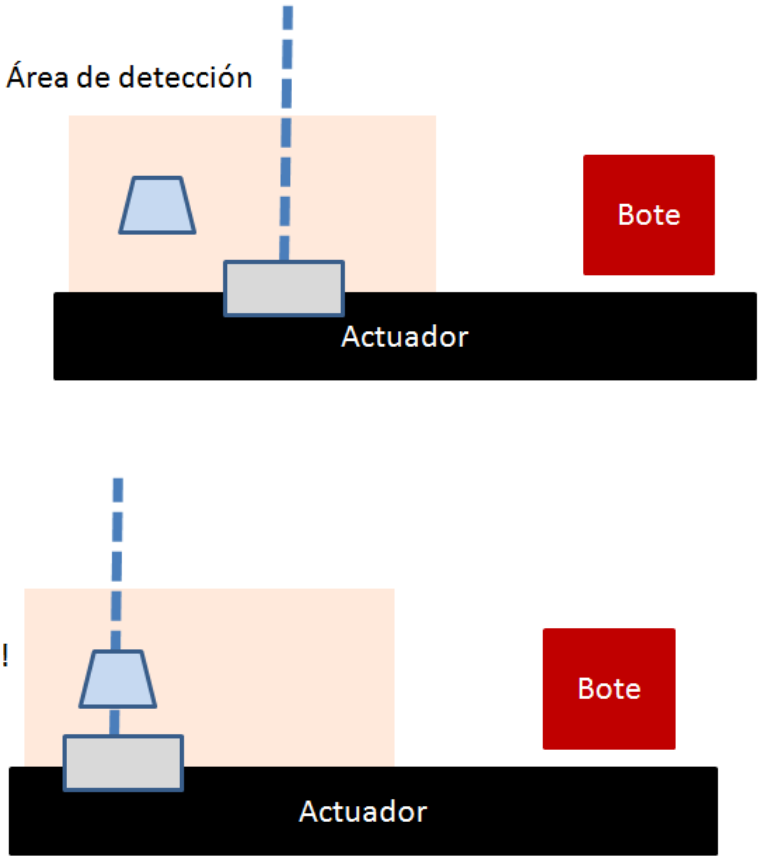
Manipulador cartesiano X Y
Requiere tener implementado el sistema de coordenadas.
Requiere sensor ultrasónico + soporte

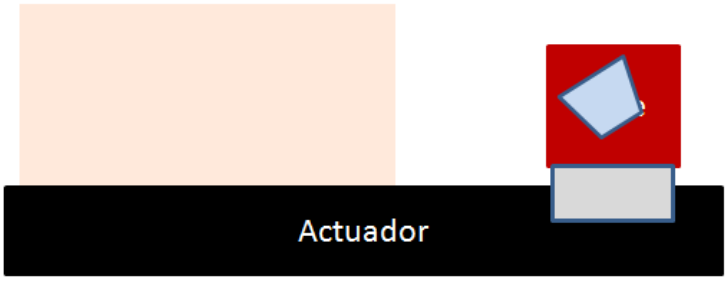
Por defecto cuando el robot se inicia, se sitúa en la posición X:20 Y:12.

El robot entrará el comportamiento/estado de "búsqueda", moviéndose de forma continua entre las coordenadas X: 0 – 20, mientras se mueve debe estar comprobado si su sensor de distancia detecta alguna caja colocada delante del robot.

En caso de que se detecte una caja el robot entrará en el comportamiento/estado "recogida" donde recogerá la caja del suelo y la depositará en un bote situado en la esquina (X:24 Y:0). Nada más depositar la caja debe volver al comportamiento inicial "búsqueda".

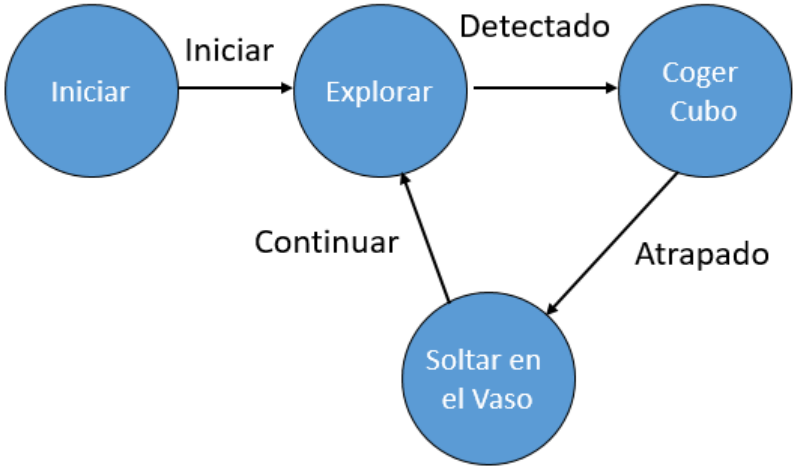
La posición del bote donde debe soltar las cajas estará pregrabada en una variable en la memoria del robot.



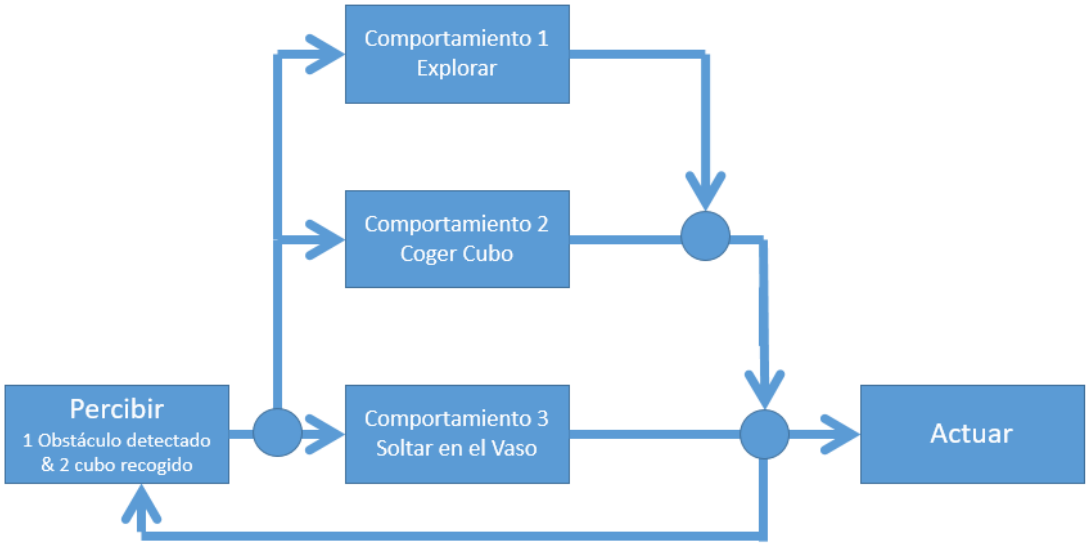


Para resolver este ejercicio se puede implementar el controlador utilizando diferentes enfoques, entre los que se encuentran:

Controlador basado en estados



Controlador basado en comportamientos



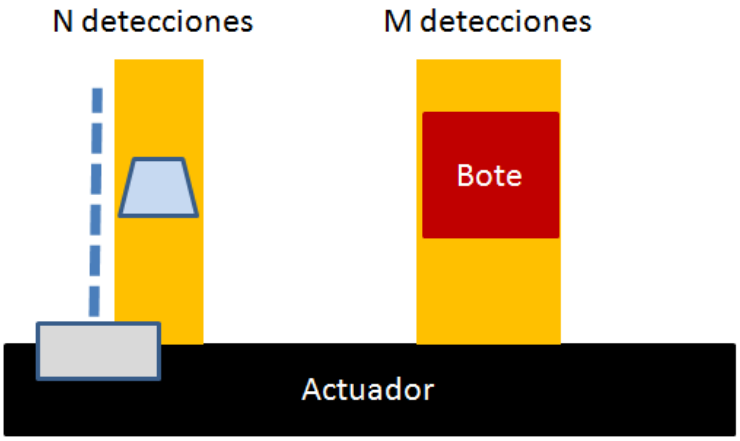
En cualquier caso, se deberá crear una función de código específica para cada estado / comportamiento.

5.7 Detección y recogida avanzada de cubo con manipulador cartesiano (1,6 Puntos)

Manipulador cartesiano X Y
Requiere tener implementado el sistema de coordenadas.
Requiere sensor ultrasónico + soporte

El objetivo de la actividad es el mismo que el de la anterior, pero el programa no requiere que se especifique la posición del bote.

En el comportamiento de "búsqueda" el robot se mueve de forma continua por todo el eje X de coordenadas (0 – 24), mientras se mueve debe estar comprobando si su sensor detecta distancia. Al detectar distancia el robot no sabe exactamente si lo que se encuentra delante es el **cubo** o el **bote**, debe seguir moviéndose y tomando detecciones, como el tamaño del cubo un número pequeño de detecciones consecutivas indicará que ha detectado el **cubo**, un número mayor de detecciones consecutivas indicará que ha detectado el **bote**.



Se debe implementar un controlador reactivo basado en comportamientos -> "búsqueda" , "detectar cubo" , "detectar bote", "coger cubo", "soltar bote en cubo". Cada comportamiento se debe implementar en una función. (Los comportamientos listados anteriormente son una orientación, si se considera más oportunos se pueden utilizar otros).

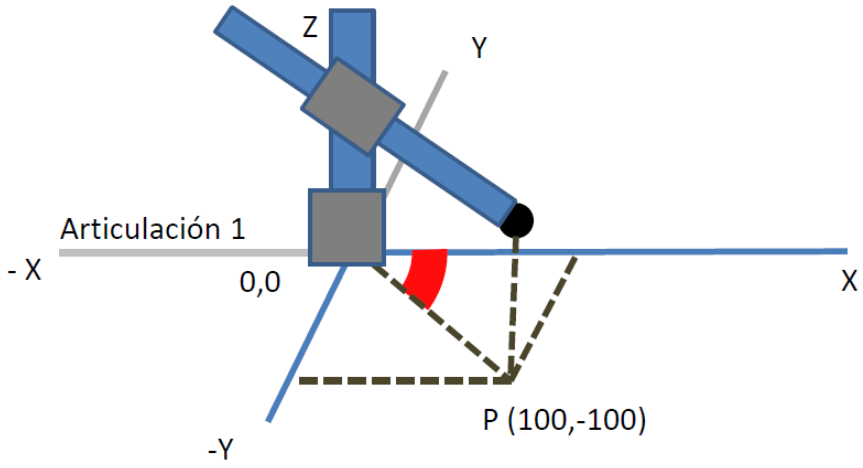
5.8 Modelo cinemático para manipulador cilíndrico (1,6 Puntos)

Manipulador cilíndrico X Y Z

El modelo cinemático permite que el robot se situé en coordenadas cartesianas X, Y, Z. El robot calcula la posición en la que debe colocar cada uno de sus servomotores para lograr situar su pinza en esa coordenada.

- Articulación 1 : Grados de giro para la base
- Articulación 2 : Coordenada para el actuador 1 eje Z
- Articulación 3 : Coordenada para el actuador 2 eje X-Y

Revisar Tema 5 pág 43 – Calculo de cinemática inversa para robot cilíndrico.



El robot debe ser capaz de recibir coordenadas X,Y,Z por consola COM.

Las coordenadas las recibirá en el formato X-Y-Z-ms (milisegundos de espera). Podrá recibir varias coordenadas separadas por , .

Como, por ejemplo:

20-0-12-3000, 0-20-12-3000, 0-20-20-3000, 12-12-20-3000

5.9 Algoritmo de planificación (1,6 Puntos)

Manipulador cilíndrico X Y Z / Manipulador cartesiano

Resolver en Java el utilizando un algoritmo de planificación (Strips) el siguiente problema de ordenación de cajas, en el cual hay 3 cajas A, B, C que se pueden situar en diferentes montones.

```
// Estado Inicial
List<Propiedad> propiedadesIni = new LinkedList<Propiedad>();
propiedadesIni.add (new Propiedad(Propiedad.CIMA, "B"));
propiedadesIni.add (new Propiedad(Propiedad.SOBRE, "B","A"));
propiedadesIni.add (new Propiedad(Propiedad.SOBRE, "A","C"));
propiedadesIni.add (new Propiedad(Propiedad.SOBRE, "C","S"));
Estado estadoInicial = new Estado(propiedadesIni);

// Estado final
List<Propiedad> objetivos = new LinkedList<Propiedad>();
objetivos.add (new Propiedad(Propiedad.CIMA, "C"));
objetivos.add (new Propiedad(Propiedad.SOBRE, "C","B"));
objetivos.add (new Propiedad(Propiedad.SOBRE, "B","A"));
objetivos.add (new Propiedad(Propiedad.SOBRE, "A","S"));
Estado estadoObjetivo = new Estado(objetivos);
```

Utilizar la plantilla de código subida al campus virtual, se requiere entregar la plantilla resuelta con un algoritmo capaz de obtener el plan (no hace falta que sea el plan más óptimo).

5.10 Entorno de emulación robot manipulador (1,6 Puntos)

Explorar el funcionamiento de los robots manipuladores en un entorno de emulación robótica.

Definir un entorno con un robot manipulador y realizar la siguiente tarea:

- Crear un programa capaz de apilar tres bloques en una torre y posteriormente desapilarlos dejándolos en su posición inicial.

Entornos de emulación robótica (libre elección).

- **RoboStudio (Recomendado)**
- V-Rep PRO EDU
- Webots
- Gazebo
- Otros (RobotStudio, Robologix, Webots, etc.).

Documentar paso a paso el proceso seguido, este apartado requiere la entrega de un documento y todos los ficheros generados desde el entorno de emulación.