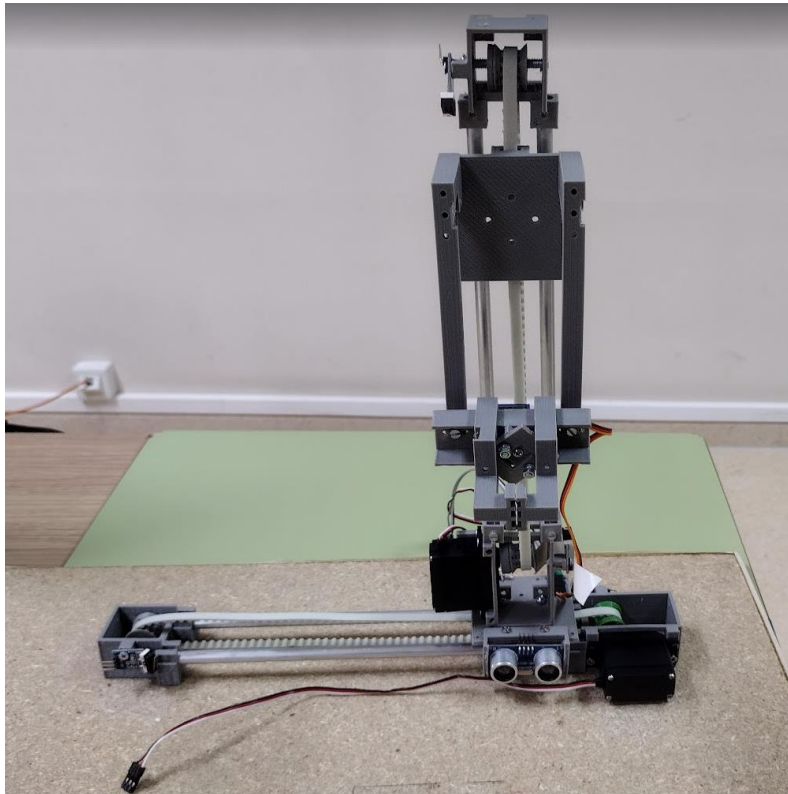


Robots manipuladores



Práctica 6 – Actividades (v1.4.2 octubre 2022)

Software para robots

Cristian González García

gonzalezcristian@uniovi.es

Basado en el material original de Jordán Pascual Espada

Índice

Actividades normales.....	2
(Recogida6.1) Detección y recogida simple de cubos con un manipulador cartesiano (1,25 Puntos)	2
(Recogida6.2) Detección y recogida avanzada de cubos con un manipulador cartesiano (1,25 Puntos)	4
Ampliaciones.....	4
(Cilíndrico6.3) Modelo cinemático para manipulador cilíndrico (1,5 Puntos).....	4
(Emulación6.4) Entorno de emulación para robot manipulador (1,25 Puntos)	5
(Diseño6.5) Diseñar en OpenSCAD un robot cartesiano (0,75 Puntos)	6
(Robot6.6) Diseñar en OpenSCAD un robot (0,75 Puntos)	6
(Simulación6.7) Entorno de emulación Mujoco (1 punto)	7

El total de las actividades tienen un valor de 2,5 puntos actividades normales y 4,25 puntos de ampliaciones dentro del **bloque 3**.

Actividades normales

(Recogida 6.1) Detección y recogida simple de cubos con un manipulador cartesiano (1,25 Puntos)

Material necesario: 2 actuadores lineales y 1 pinza (montados formarán el cartesiano XY), 4 sensores de colisión, 2 joysticks, 1 sensor de ultrasonidos (soporte en un actuador), 1 vaso y 3 cubos/objetos.

Requiere tener implementado el ejercicio opcional 4.4 y el 5.2

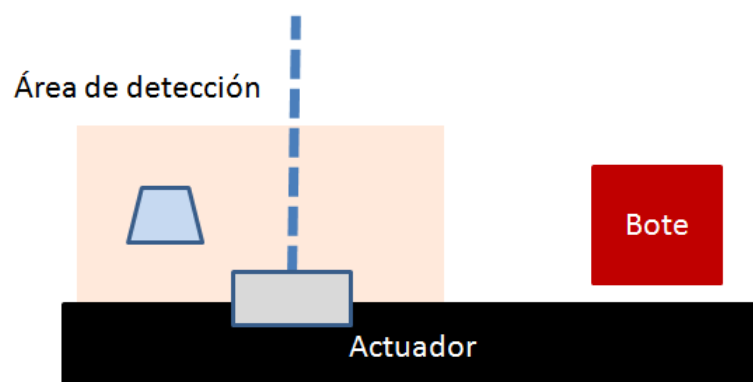
Por defecto, cuando el robot se inicia, se sitúa en la posición X:20 Y:12.

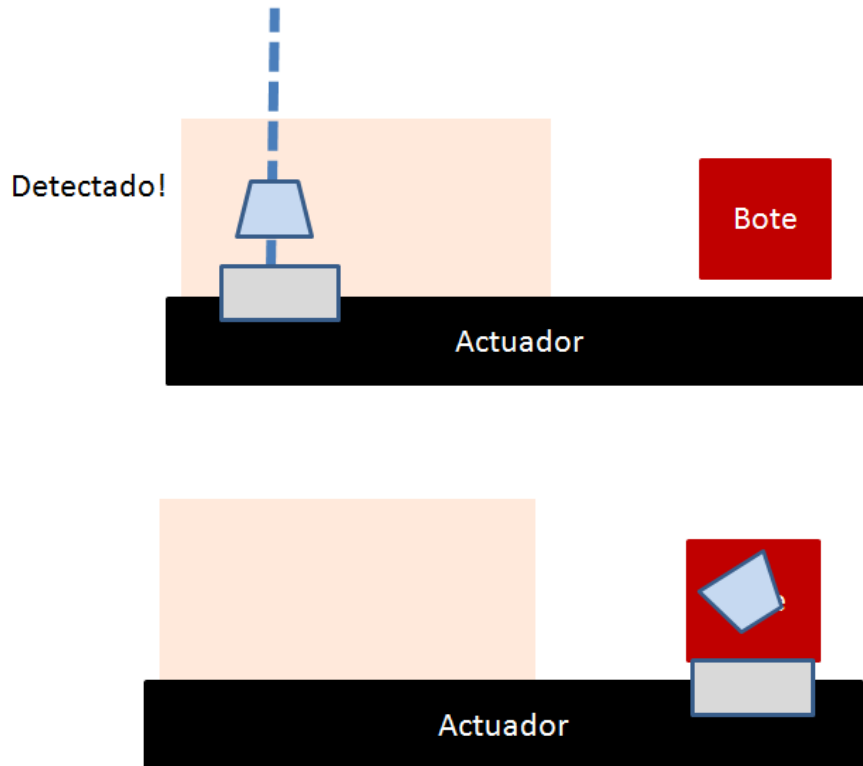
El robot entrará en el modo/comportamiento/estado de «búsqueda», moviéndose de forma continua entre las coordenadas X: 0 – 20. Mientras se mueve, debe estar comprobando si su sensor de distancia detecta alguna caja colocada delante del robot.

En caso de que se detecte una caja, el robot entrará en el modo/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».

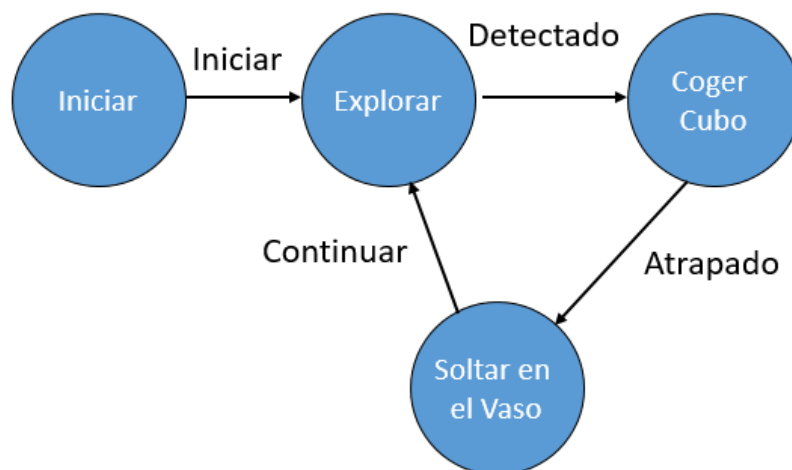
Para probar esto, el robot deberá moverse de izquierda a derecha y entonces se le tendrá que poner una caja en algún punto, el robot la detecte y la recoja. Se le ponga otra caja en otro sitio, la detecte y la recoja. Así continuamente.

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





Para resolver este ejercicio, se puede seguir el siguiente esquema de estados:



Se deberá crear una función de código específica para cada modo/estado/comportamiento.

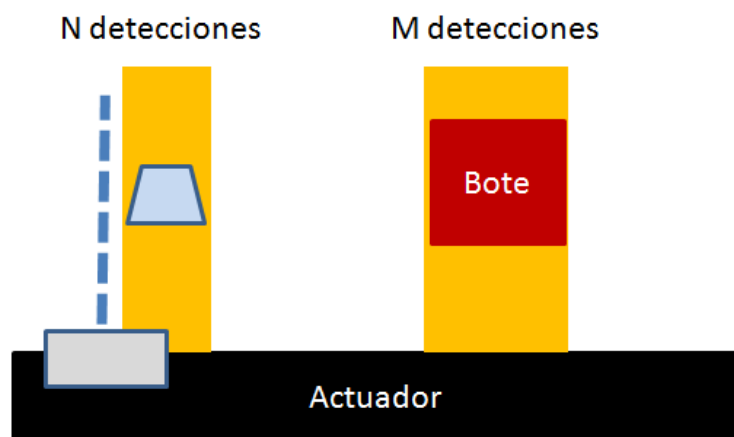
(Recogida6.2) Detección y recogida avanzada de cubos con un manipulador cartesiano (1,25 Puntos)

Material necesario: 2 actuadores lineales y 1 pinza (montados formarán el cartesiano XY), 4 sensores de colisión, 2 joysticks, 1 sensor de ultrasonidos (soporte en un actuador), 1 vaso y 3 cubos/objetos.

Requiere tener implementado el ejercicio opcional 4.4, el 5.2 y el obligatorio 6.1

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

En el modo de «búsqueda», el robot se moverá de forma continuada por todo el eje X de coordenadas (0 – 24). Mientras se mueve, el robot debe estar comprobando si su sensor detecta algo dentro de su distancia. Al detectar un objeto, debido a que el robot no sabrá exactamente si lo que se ha encontrado delante es el **cubo** o el **bote**, debe seguir moviéndose y tomando detecciones, como, por ejemplo, el tamaño del cubo y el bote. Esto se puede conseguir mediante un número pequeño de detecciones consecutivas que indicarán que se ha detectado un **cubo**. En cambio, si hay un número mayor de detecciones consecutivas indicará que ha detectado el **bote**.



Se debe implementar un control 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 oportuno se pueden utilizar otros.

Ampliaciones

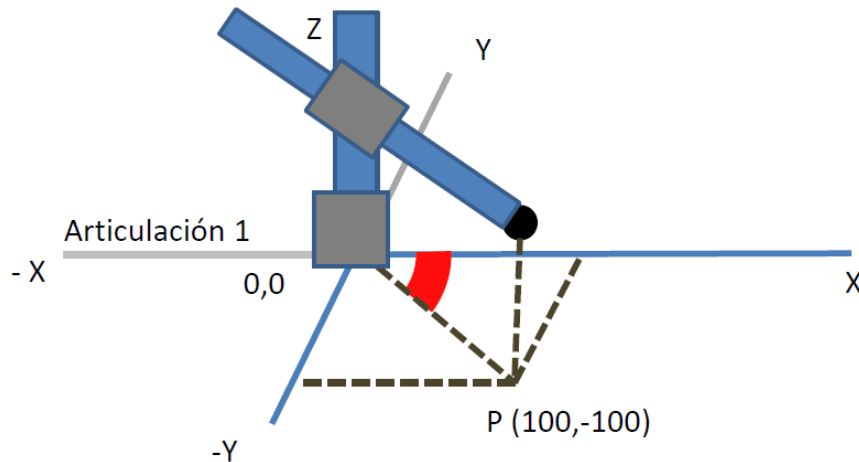
(Cilíndrico6.3) Modelo cinemático para manipulador cilíndrico (1,5 Puntos)

Material necesario: 1 brazo robótico + complementos.

El modelo cinemático permite que el robot se sitúe en las 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 diapositivas del tema 5 – 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 «;». Es el mismo sistema utilizado para 4.5.

Como, por ejemplo:

20,0,12,3000;0,20,12,3000;0,20,20,3000;12,12,20,3000;

Recordad que no hay función Split y que tendréis que haceros vuestro propio Split (posibilidad usando método substring), o utilizar la función «strtok» o «strchr» de C++.

(Emulación6.4) Entorno de emulación para robot manipulador (1,25 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:

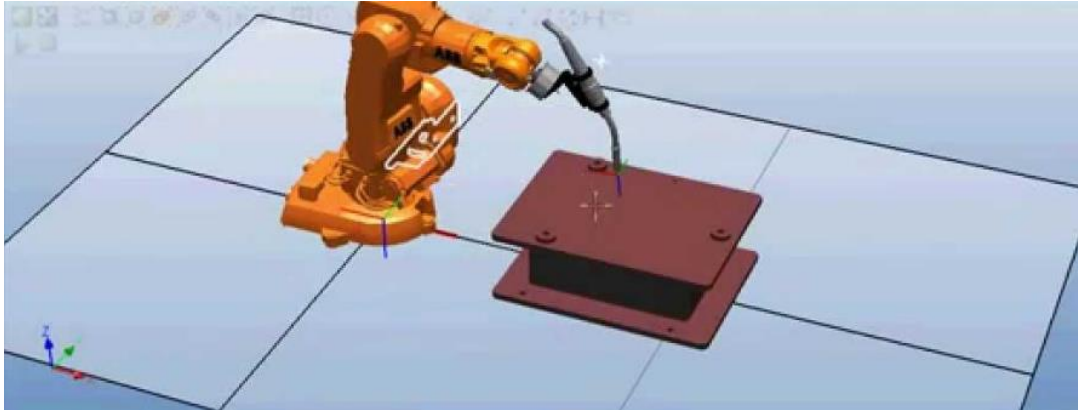
- Crear un programa capaz de apilar tres bloques en una torre y posteriormente desapilarlos dejándolos en su posición inicial.

Entornos de emulación robótica (libre elección):

- **RobotStudio (Recomendado)**
 - Ayuda: <https://www.youtube.com/watch?v=RSjCAmaS8ac>

- V-Rep PRO EDU
- Webots
- Gazebo
- Otros (Robologix, Webots, etc.).

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



(Diseño6.5) Diseñar en OpenSCAD un robot cartesiano (0,75 Puntos)

Utilizando OpenSCAD, hay que diseñar un actuador lineal que permita, mediante el uso de varios de estos, crear diferentes robots cartesianos.

Hay que tener en cuenta las medidas de los motores, su acople al modelo, y el acoplamiento de varios actuadores lineales para construir el robot cartesiano.

Los mecanismos de movimiento pueden ser cualquiera de los explicados en las diapositivas tituladas «Tipos de movimientos y mecanismos, y consejos de su modelado» y que se encuentran en el Campus Virtual, así como el uso de librerías que ayude a crearlo.

El actuador lineal tendrá que estar dividido en diferentes archivos, 1 por pieza. Se juntarán todos en un archivo para crear el actuador lineal. Después, los diferentes robots cartesianos importarán el archivo del actuador lineal y se montarán para crear un robot cartesiano por fichero.

Las normas son las mismas que en la práctica de OpenSCAD.

(Robot6.6) Diseñar en OpenSCAD un robot (0,75 Puntos)

En este ejercicio, de temática libre, hay que diseñar un robot que utilice los diferentes mecanismos de transmisión de movimientos en las diapositivas tituladas «Tipos de movimientos y mecanismos, y consejos de su modelado» y que se encuentran en el Campus Virtual. Todos estos mecanismos se diseñarán en OpenSCAD, ya sea con librerías o de forma manual.

Las diferentes partes del robot deberán ir divididas en piezas y se juntarán en un fichero «main».

Las normas son las mismas que en la práctica de OpenSCAD.

(Simulación6.7) Entorno de emulación Mujoco (1 punto)

En este ejercicio se trata de investigar el entorno de emulación Mujoco (<https://mujoco.org/> y <https://deepmind.com/blog/announcements/mujoco>) y buscar, modificar pequeño ejemplo que exista para trabajar con él y ver el funcionamiento del entorno.

Además del vídeo, en este caso, hay que añadir las webs utilizadas y la descarga de los recursos, así como dónde está su uso en el ejemplo en el PDF a entregar.