

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 (prácticas 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.

5.1 Grabación de tareas sobre un actuador lineal (1,6 Punto)

Actuador lineal simple

Utilizar el sistema de teleoperación implementado anteriormente para permitir grabar tareas. Una tecla del PC debe servir para activar / detener el modo grabación.

Una vez comience el modo grabación el usuario mueve el actuador lineal de forma libre (utilizando el joystick), todas las acciones que va realizando quedan registradas, hasta que se vuelve a pulsar la tecla de activar/detener el modo grabación.

Otra tecla del PC debe servir para reproducir de forma automática la última tarea grabada.

El actuador solo almacena una tarea, la última que se ha grabado.

Cuando el modo de grabación de tarea está activo el motor del actuador se debería mover más despacio de lo normal, de esta forma el operario podrá realizar la tarea de forma más precisa.

Para grabar una tarea hace falta saber, que acción se está realizando: ALANTE, ATRAS, (en que coordenada se inicia el movimiento y en cual se finaliza), para las pausas PARADO hace falta saber cuánto tiempo se han mantenido.

Se puede/debe establecer un máximo de instrucciones que serán grabadas, por ejemplo 100.

5.2 Manipulador cartesiano teleoperado (1,6 Punto)

Manipulador cartesiano X Y

Construir un robot manipulador cartesiano utilizando dos actuadores lineales y una pinza.

Un **joystick 1** debe controlar el movimiento del carro del **actuador eje X y del actuador eje Y**.

El **joystick 2** debe controlar el movimiento de la pinza. Al mover el **joystick 2** hacia la izquierda la pinza debe comenzar a cerrarse lentamente, al mover el **joystick 2** hacia la derecha la pinza debe comenzar a abrirse lentamente, si en cualquier momento el usuario suelta el **joystick 2** la pinza debe dejar de moverse y permanecer en el estado actual.

Para controlar el servo de 180 grados, debemos almacenar sus grados actuales, en el caso de que se quiera abrir la pinza vamos sumando grados poco a poco a sus grados actuales, en el caso de que se quiera cerrar la pinza vamos restando grados poco a poco a su estado actual.

Ver ejemplo del código para controlar la pinza en el campus virtual

5.3 Grabación de tareas en manipulador cartesiano (1,6 Punto)

Manipulador cartesiano X Y

Ampliar la actividad "5.1 Grabación de tareas sobre un actuador lineal" para que funcionen en manipulador cartesiano X, Y.

- La grabación de tareas debe registrar el movimiento de los dos actuadores **y de la pinza**.

El robot solo almacena una tarea, la ultima que se ha grabado.

5.4 Apilar cajas con manipulador cartesiano (1,6 Punto)

Manipulador cartesiano X Y

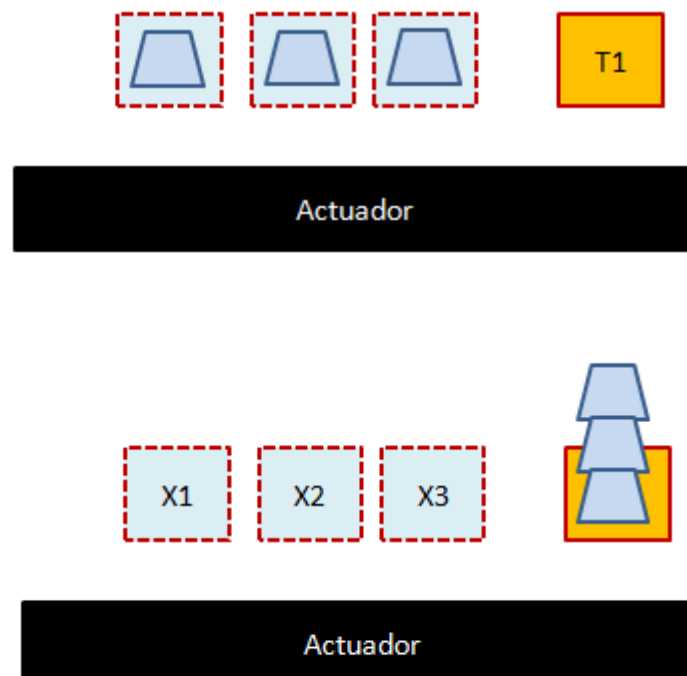
Desarrollar un programa que tenga pregrabadas en variables las coordenadas donde se encuentran **dos** cajas y las coordenadas del punto donde debe depositarlas en forma de torre.

El programa debe hacer que el robot manipulador cartesiano recoja cada una de las dos cajas y las vaya apilando en el punto indicado, debe apilar las cajas unas encima de otras (sin que se caigan).

Se debe ir calculando internamente la altura de la torre de cajas para que la pinza no tire la ninguna de las cajas por error.

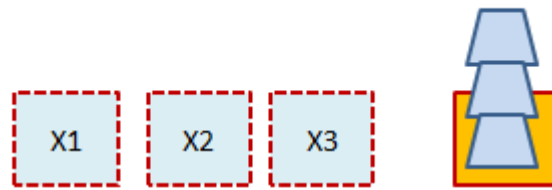
Para que el sistema de coordenadas funcione de forma correcta, debemos saber el tiempo que tarda el robot en ir de una esquina hacia la otra y dividirlo entre 24 (estamos suponiendo que el sistema tiene un eje de coordenadas que va desde 0 hasta 24).

La forma más correcta de obtener ese tiempo es hacer que el robot se inicie en modo “calibración” cada vez que se activa la aplicación y calcule el tiempo de movimiento que necesita para moverse una coordenada (al igual que se hizo en la actividad 4.3) .



Ampliación opcional (no puntúa)

Desapilar las cajas apiladas y volver al estado inicial.



Actuador



Actuador

5.5 Instrucciones de alto nivel (1,6 Puntos)

Manipulador cartesiano

Crear un intérprete de instrucciones de alto nivel que puedan ser enviadas al robot manipulador a través del COM.

MOV_{XY}(X,Y) - se mueve hacia un punto, primero avanzando en el eje Y (hasta llegar a la posición) y luego en el eje X.

MOV_{PX}(X,Y) - se mueve hacia un punto, primero avanzando en el eje X (hasta llegar a la posición) y luego en el eje Y.

APINZA - abre la pinza, los grados aplicados al motor para abrir la pinza son conocidos por el robot.

CPINZA - cierra la pinza, los grados aplicados al motor para abrir la pinza son conocidos por el motor, no tiene que cerrarse completamente, pero sí lo necesario para coger un cubo.

WAIT(milisegundos) – realiza una pausa del número de milisegundos indicado.

Los comandos se deben poder introducir por el puerto COM, separados por ;

Ejemplo de conjunto de instrucciones enviadas al robot:

APINZA;MOV_{XY}(10,20); MOV_{XY}(10,0);CPINZA;WAIT(1000); MOV_{PX}(24,24)