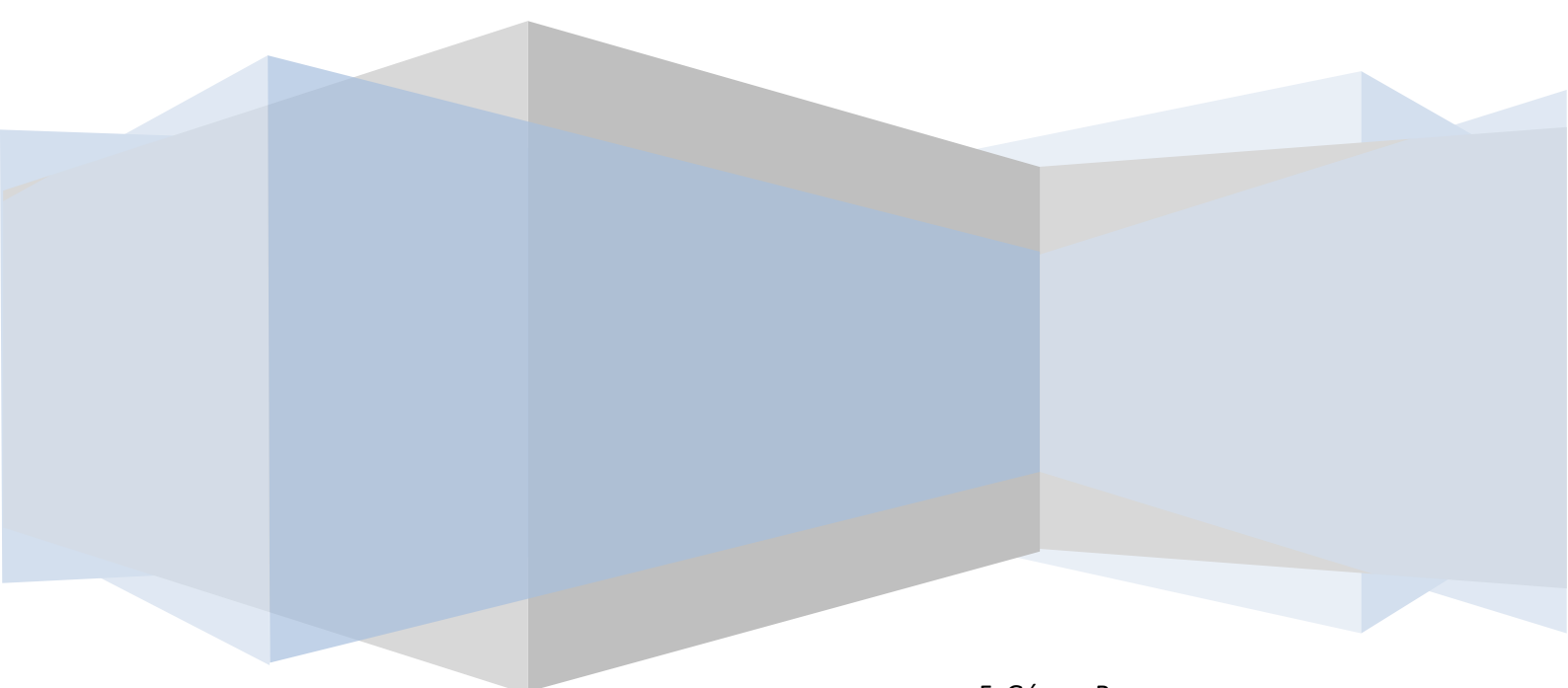




Práctica 2.1

Control de movimientos Simulados de un
Manipulador de 6 Grados de Libertad.



ENUNCIADO

El objetivo de esta práctica es familiarizarse con las estrategias de programación de manipuladores. Para ello vamos a utilizar los programas **Matlab** y **RoboDK**.

Al ejecutar el programa RoboDK, aparecerá la pantalla vacía que se ve en la figura 1 a).

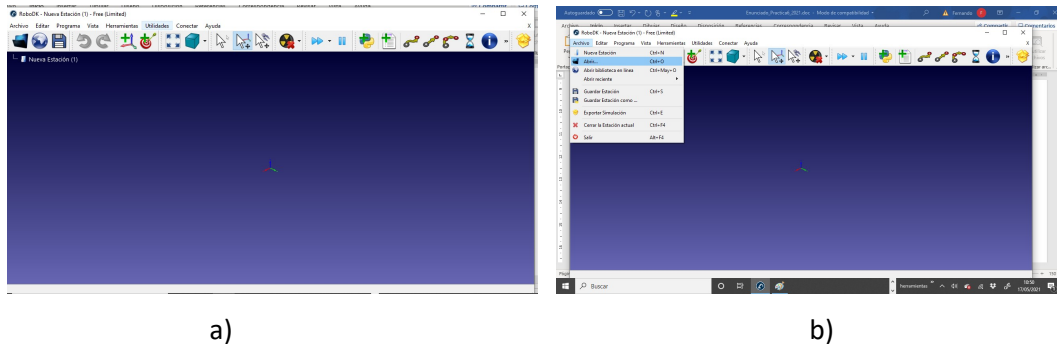


Figura 1

Posteriormente seleccionaremos la pestaña abrir (ver figura 1 b) y buscaremos en nuestra estructura de directorio el fichero 'Estacion_Lab_6_2'. Al seleccionar este archivo aparecerá un escenario con un robot UR3 y tres bloques como los que se ven en la figura 2.

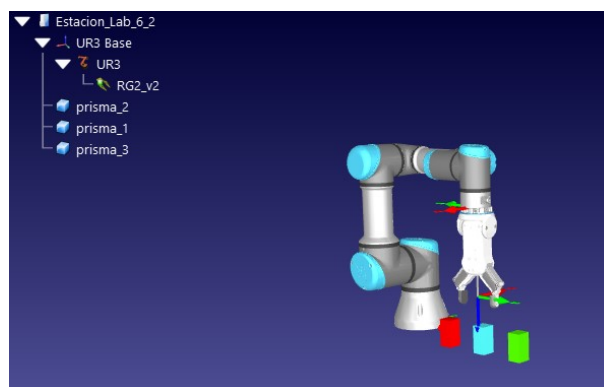


Figura 2

Se pretende que el alumno realice la planificación articular de una tarea de 'pick and place' y posteriormente realice un programa de manera que el manipulador ejecute los movimientos previstos. Para ello se dispone de varias funciones compiladas que permiten simular el movimiento del manipulador en RoboDK mediante comandos en Matlab.

La primera es la función `Iniciacion()` que puede llamarse de la forma:

```
codigo=1;
Identificador = Iniciacion('RBDK',codigo);
```

Si se ejecuta este comando se establece una conexión entre Matlab y RoboDK. Si todo transcurre normalmente, Matlab debe devolver:

```
Identificador =
3
```

El valor que tome la variable `codigo` es indiferente mientras trabajemos con RoboDK. La variable `Identificador` la utilizaremos en la llamada al resto de funciones para identificar la comunicación entre Matlab y RoboDK. Para poder controlar la simulación desde Matlab es imprescindible haber ejecutado este comando y utilizar correctamente la variable 'Identificador'.

La segunda función es `MoveJ_Robot_lab` cuyo prototipo es:

```
MoveJ_Robot_lab(angulos, aceleracion, velocidad, Identificador, codigo)
```

Permite mover las articulaciones del manipulador de acuerdo con los valores configurados en la variable `angulos` que será un vector con seis componentes. Estas 6 componentes se corresponderán con los ángulos hacia los que se moverán las articulaciones y su valor se expresará en radianes. El valor de la variable `velocidad` determina la velocidad con la que giran las articulaciones, la variable `aceleración` determina el valor con el que se acelera el movimiento de las articulaciones. En cambio, el valor de la variable `codigo` solo afectará cuando trabajemos con el robot real, no tienen ningún efecto en las simulaciones efectuadas en RoboDK, por tanto, en este ejemplo, haremos que tome cualquier valor de dos cifras.

Si se ejecuta el comando:

```
MoveJ_Robot_lab([0 -3*pi/4 -pi/2 0 0 0], 1, 1, Identificador, 1)
```

el manipulador se moverá hasta la configuración mostrada en la figura 3.

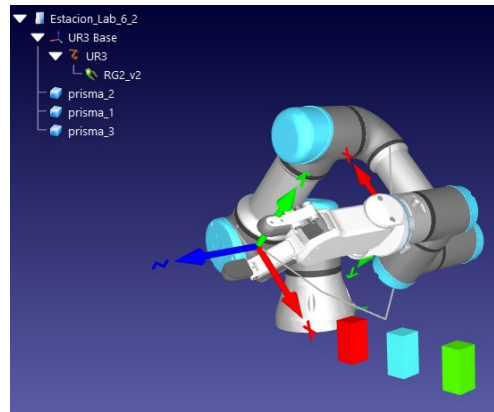


Figura 3

La llamada a la función `'Ready_lab(Identificador,codigo)'` hace que el manipulador vuelva a la posición inicial (Home) en la que estaba cuando se abrió el fichero `'Estacion_Lab_6_2.rdk'`.

La función `'RG2_lab(100,Identificador, codigo)'` permite accionar la pinza. La pinza agarra la pieza más cercana si el valor de `abertura` es menor que 100 y deja de agarrarla si el valor de `abertura` es mayor o igual a 100.

Si la pinza se coloca abierta y correctamente sobre una pieza, y se invoca la función para que la pinza agarre un bloque, el manipulador podrá transportar la pieza modificando su posición tal y como se ve en la figura 4.

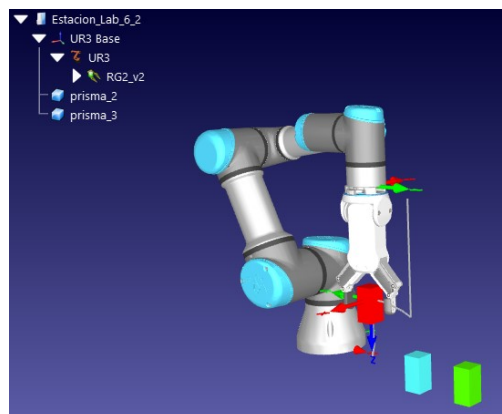


Figura 4.

Cuando trabajemos con el robot real esta variable tomará valores consecuentes con el ancho de la pieza a transportar.

Para manipular correctamente una pieza, primero es necesario mover el manipulador de manera que la pinza se sitúe en una CONFIGURACIÓN DE APROXIMACIÓN. La configuración de aproximación suele determinarse a partir de la CONFIGURACIÓN DE MANIPULACIÓN, pero con la pinza desplazada una distancia (que dependerá de cada situación) a lo largo del eje Z del SISTEMA DE REFERENCIA DE LA HERRAMIENTA. En la figura 5 a) se muestra la configuración de aproximación y en la figura 5 b) la configuración de manipulación. Esta última con la pinza colocada correctamente sobre la pieza.

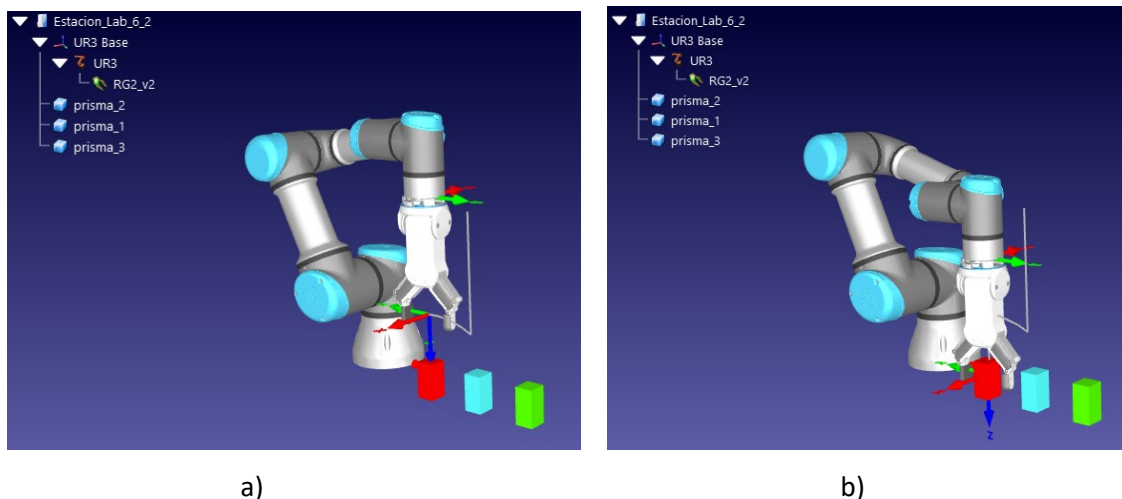


Figura 5

Si en ese momento se actuara la pinza, y se ejecuta un movimiento de retorno a la configuración de aproximación el sistema evolucionaría a la situación representada en la figura 4.

RoboDK es un programa completo que facilita la programación *offline* de manipuladores industriales. En la clase, se darán explicaciones sobre la interpretación de la información que ofrece el interfaz de éste.

RoboDK tiene su propio lenguaje, que permite incluir rutinas en Python y simular numerosos tipos de manipuladores. Es posible acceder a un manual detallado de instalación y uso de recursos básicos en el enlace:

<https://robodk.com/doc/es/Basic-Guide.html#Start>

EJERCICIO

- Realice varios movimientos de las articulaciones del manipulador para familiarizarse con el comportamiento del simulador.
- Utilizando lo aprendido en las prácticas anteriores determinar una serie de configuraciones que permitan recoger las piezas del escenario definido en el fichero 'Estacion_Lab_6_2.rdk' y colocarlas según se ve en la siguiente figura.

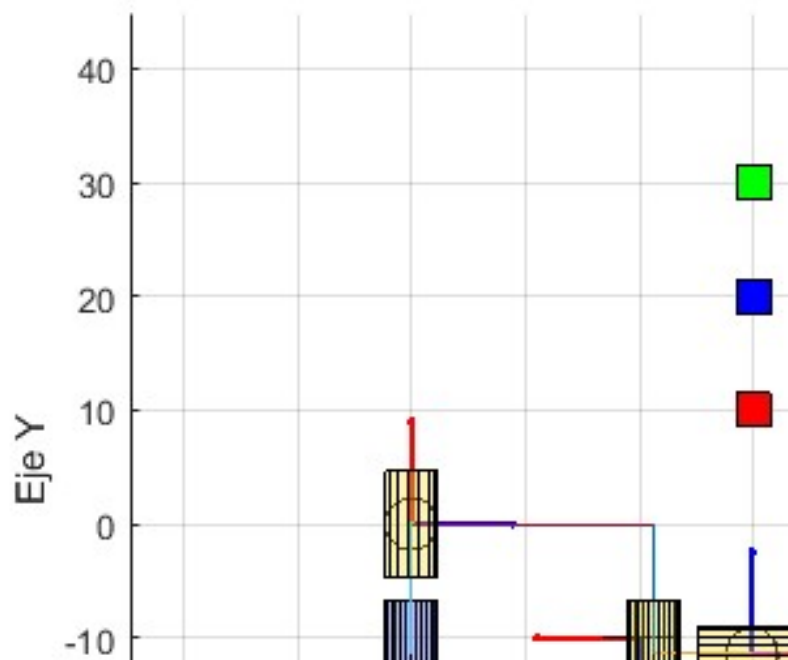


Figura 6

- Trate de realizar el código para que sea lo más eficiente posible, reduciendo el número de configuraciones calculadas.