

# PROGRAMACIÓN DE ROBOTS

## *Robots Manipuladores* *Tutorial de programación en Matlab*

El objetivo del tutorial es facilitar la realización de las prácticas de programación de robots manipuladores usando la librería Matlab para Scorbob.

### **1.- Recordatorio antes de comenzar a trabajar**

Antes de comenzar a trabajar con el robot, hay que hacer dos cosas:

- Seleccionar como directorio actual la carpeta [Y:\](#) (esta carpeta será siempre la carpeta de trabajo, así que hay que seleccionarla SIEMPRE que se use el robot)
- Descargar y descomprimir en dicha carpeta los ficheros relativos al manejo del Scorbob con Matlab que están colgados en la web de la asignatura. Una vez descomprimida en vuestro directorio de trabajo, ya no es necesario volverla a descomprimir.
- Añadir al path nuestra carpeta [Y:/](#) (y todas sus subcarpetas)

### **2.- Creación de programas**

Los programas para realizar las prácticas se crean usando el editor de programas incluido en Matlab. Cosas a tener en cuenta:

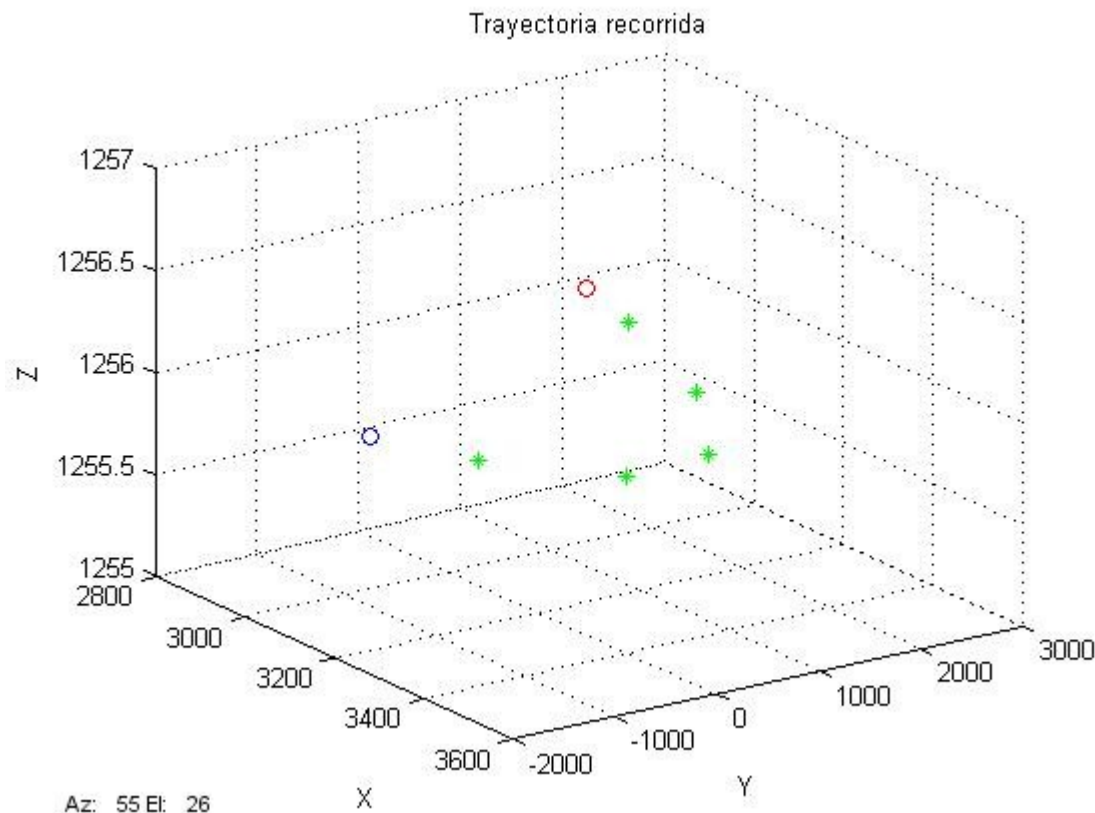
- Para obtener ayuda de cualquier función Matlab, basta con teclear *help nombre\_función*.
- Para obtener ayuda específica de la librería de programación del Scorbob, teclear *help Scorbob*.
- Los programas se guardan con la extensión .m
- Para ejecutar un trozo de código escrito en el programa, seleccionar dicho código, pinchar con el botón derecho del ratón y escoger la opción de ejecución de código, o bien pulsar F9.
- Para detener un programa, usar Ctrl-C. ¡¡Cuidado, eso puede afectar al uso de las funciones de la librería Matlab para Scorbob!!
  - ¡¡ALGUIEN DEBE ESTAR CERCA DEL BOTÓN DE PARADA DEL SCORBOT!!
- Los comentarios en Matlab se crean comenzando con %. Es conveniente comentar el código adecuadamente.

### **3.- Recordatorio: programación básica en Matlab**

1. Estructuras de programación: *if*, *for*, *while*. Para ver la sintaxis Matlab, hacer *help estructura* en la ventana de comandos.
2. Matrices
  - En Matlab, todas las variables son matrices.
  - Por ejemplo, una matriz se define de esta manera: `m = [a11 a12 a13; a21 a22 a23; a31 a32 a33]`
  - Las posiciones de las matrices se acceden usando paréntesis; por ejemplo, para el caso anterior: `ele = m(2,3);`
  - Los índices en Matlab comienzan en 1.
3. Dibujo
  - Para dibujar las posiciones, usar la instrucción *plot*. Recordad que la ventana en la

que se abre la figura ofrece la posibilidad de moverla en 3D.

- Un gráfico debe incluir títulos, etiquetas de los ejes, y leyenda. Estas son las instrucciones Matlab correspondientes: *title*, *xlabel*, *ylabel*, *legend*; si se quiere añadir una rejilla, usar *grid*.
- Para añadir elementos a una gráfica ya existente, usar, antes del *plot*, el comando *hold on* (basta con usarlo una vez).
- Para abrir una nueva ventana gráfica, usar *figure*.
- Las imágenes pueden guardarse/exportarse usando el menú.
- Por ejemplo, este sería el aspecto de una gráfica que muestra las posiciones de una trayectoria recorrida por el robot:



4. Echo: las instrucciones en Matlab pueden finalizarse sin ; o con ;
  - Sin ; se realiza eco (*echo*), es decir, se muestra el resultado de la instrucción por pantalla; es útil para depurar, pero enlentece y dificulta la visión del programa.
  - Con ; no hay eco y no se muestran los resultados por pantalla.
5. Impresión de mensajes
  - Para imprimir mensajes por pantalla, usar la instrucción *fprintf* (similar a la de C); las cadenas se escriben entre comillas simples.
  - Para esperar a que el usuario introduzca un carácter por pantalla, usar *input*.

#### **4.- Programación del Scorbob usando la librería (modo real)**

IMPORTANTE:



- USAR ÚNICAMENTE LAS FUNCIONES QUE COMIENZAN POR *Scb*
- ¡¡ALGUIEN DEBE ESTAR CERCA DEL BOTÓN DE PARADA DEL SCORBOT!!

Aspectos importantes a tener en cuenta:

- El programa debe inicializarse y finalizarse de manera civilizada (ver la plantilla *plantillaScorbotMatlab\_real.m* en la web)
- Al establecer la velocidad del brazo (*ScbSpeed*), NUNCA usar un valor superior a 20.
- Las posiciones que maneja la librería Matlab para acceder al Scorbot son estructuras con diferentes campos; las coordenadas cartesianas se encuentran en el campo *xyz*. Por tanto, para posición que hayamos denominado *ap1*, en *ap1.xyz(1)* tendríamos la coordenada x, en *ap1.xyz(2)* estaría la coordenada y, y en *ap1.xyz(3)* se almacenaría la coordenada z.
- La instrucción de movimiento *ScbMove* debe ser bloqueante, es decir, asegurarse de que finaliza el movimiento actual antes de comenzar el siguiente.
- Si la instrucción *ScbMove* devuelve las posiciones recorridas, lo hace en una estructura celda de Matlab (una matriz de matrices). Para poder acceder a cada posición individualmente (por ejemplo, para dibujar la trayectoria), la sintaxis es la siguiente: *p{i}*, donde *i* es el índice de la posición; por ejemplo, *p{i}.xyz(1)* contendría la coordenada x de la posición *{i}*.
- Si se van a grabar las posiciones almacenadas en el espacio de trabajo, guardar únicamente las posiciones: es decir, hay que asegurarse de que el objeto robot definido por la función *ScbHome* no está incluido en el fichero .mat que se grabe.

## **5.- Programación del Scorbot usando v-rep (modo simulación)**

En primer lugar, ejecutar V-Rep, y cargar la escena principal (*main\_scene.ttt*) incluida en el fichero comprimido disponible en la web de la asignatura. Pasos a seguir:

- Pulsar el botón *Play*.
- Para trasladar las piezas que aparecen en la escena, usar el botón  de la barra de opciones. Para cambiar su posición, abrir la pestaña *Position*.
- Para rotar las piezas que aparecen en la escena, usar el botón  de la barra de opciones; la rotación se realiza sobre el eje Y.

Lanzar Matlab, teniendo en cuenta estos aspectos además de todos los mencionados para el modo real:

- El programa debe inicializarse y finalizarse de manera civilizada (ver la plantilla *plantillaScorbotMatlab\_vrep.m* en la web).
- En simulación, no dibujar las posiciones con *plot*.
- Las posiciones a las que se irá moviendo el robot pueden cargarse desde un fichero .mat con las posiciones previamente cargadas (opción *teach=0*, ver la plantilla mencionada anteriormente), o bien guardarlas usando la pistola de programación implantada en Matlab (opción *teach=1*).
- Las posiciones grabadas en el modo real y las posiciones grabadas en el modo simulado no son compatibles: es decir, no pueden usarse las posiciones reales grabadas en el lab con el robot de simulación.