



INSTITUTO DE INNOVACIÓN Y TECNOLOGÍA APLICADA

M.E – Rotación – Repaso



Máquina de Estados

Una máquina de estados se trata de una estructura la cual permitirá controlar el flujo de nuestro controlador.

En este sentido, definiremos estados de tal manera que mientras nos encontremos en uno en concreto, se ejecute solo una cantidad definida de acciones.



Máquina de Estados - Aplicación

Para aplicar una máquina de estados, empezaremos definiendo una variable ESTADOS. Dicha variable, alternará sus valores entre los estados que definamos.

Es importante recordar, que esta variable cambiará continuamente al cambiar de estado a estado.



Máquina de Estados - Aplicación

Por otro lado, utilizaremos estructuras de control (If - Elif - Else) para definir las instrucciones ordenadamente, dependiendo del estado que se le asigne a la variable ESTADO.

Nótese que podemos definir muchos estados, según el problema que deseamos resolver y sus condiciones.



Máquina de Estados - Pasos

1. Definir la cantidad de estados: Debemos conocer cuantos estados necesitamos para resolver el problema.
2. Asignar un nombre: Cada estado debe tener un nombre.
3. Crear un esquema: Escribir un esquema nos puede ayudar a interpretar para qué servirá cada estado, y en que momento es necesario aplicarlo.



Máquina de Estados - Programación

Para programar una máquina de estados en Python, crearemos una variable ESTADO y le asignaremos el nombre del estado con el que queremos que comience el ciclo infinito (cuerpo principal).

Consecuentemente, en el cuerpo principal incluiremos una cláusula if, tal que condicione en que estado nos encontramos y que acciones debe ejecutar ahí.



Máquina de Estados - Programación

```
# Programa de simulación de los estados de un semáforo
```

```
import time as tm
```

```
state = 'red' → Defino mi variable y mi primer estado
```

```
count = 0 → Agrego un contador
```

```
8 while count <= 3: → Ciclo while (Ejecución principal)
```

```
9     if state == 'red': → ¿Me encuentro en tal estado?
```

```
10         print(f'El semáforo está en {state}')
```

```
11         tm.sleep(3)
```

```
12         state = 'yellow'
```

} Ejecuto y
cambio de
Estado



Máquina de Estados - Programación

¡Un estado debe ser finito! No podemos establecer un estado y no cambiar nunca a otro, dado a que sería una aplicación inútil y podría realizarse sin aplicar estados.

Cuando usamos máquinas de Estados, las estructuras de control nos ayudarán a desplazarnos entre estados.



Rotación del Robot



Tipos de Rotación

Existen dos maneras de rotar nuestro robot:

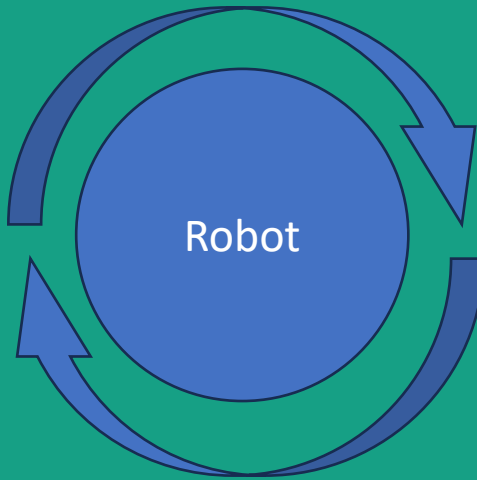
- Sobre su propio eje

- Sobre un solo eje fijo

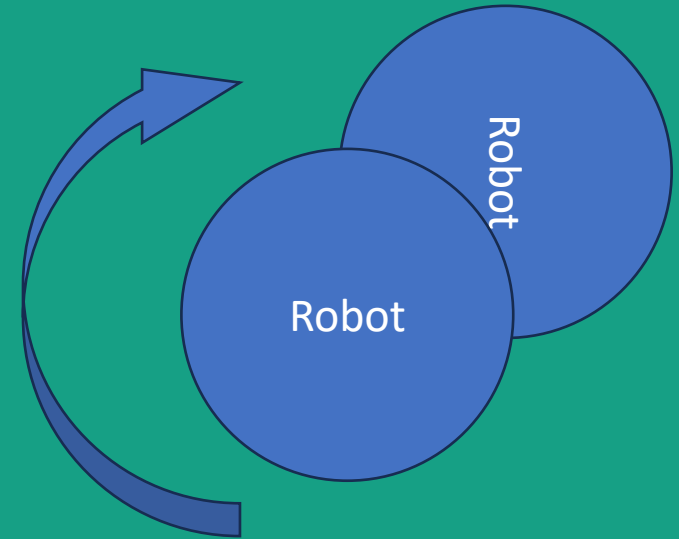
En el primer caso, el robot rotará en la misma posición que se encuentre, mientras que en el segundo cambiará su posición (o eje de origen) respecto a una dos ruedas.



Rotación - Gráficas



Rotación sobre el
propio Eje



Rotación sobre
un solo Eje



Rotación del Robot

Por conveniencia, la rotación la realizaremos sobre el propio eje del robot dado el espacio reducido de los caminos.

Dentro de la simulación, podemos lograr que rote utilizando dos técnicas:

- Rotación por Encoder
- Rotación por sensor Gyro (Giróscopo)



Rotación del Robot - Encoder

El Encoder se trata de un sensor de posición que se ubica en las ruedas del robot. Mediante cálculos algebraicos podemos calcular la rotación de las ruedas tal que el robot gire los grados que necesitamos.

Recordemos que el simulador trabaja en radianes.



Rotación del Robot - Gyro

El sensor Gyro o Giróscopio permite conocer la orientación del robot. El utilizarlo nos simplificará la tarea de la rotación, ya que permite calcularla de manera que cada giro es preciso y se obtiene un desajuste mínimo.



Sensor de Distancia



Sensor de Distancia

El sensor de Distancia es utilizado para medir una distancia entre una pared y un objeto. Su utilización es proporcional al cambio de la estructuración del mapa, por lo que se convierte en una herramienta precisa para evitar colisionar contra objetos y definir la trayectoria del robot.



Sintaxis del Programa



EL PROGRAMA

Como hemos visto hasta ahora, el programa o controlador Python suele tener un patrón en la codificación. A continuación se detallará la secuencia para definir un controlador:



Armar nuestro controlador

- Importar librerías: Por defecto importaremos controller y sus correspondientes objetos (GPS, DistanceSensor, Robot, todas relacionadas a partes del robot)
- Definir el robot: El comando Robot() creará una referencia escrita que será interpretada por el simulador. Es crear un robot con características predefinidas.
- Declarar componentes: Obtenemos los objetos que deseamos añadir con 'componente'.getDevice('etiqueta'). La etiqueta debe coincidir con la que utiliza el archivo .json del robot.
- Activar componentes: Con el comando 'componente'.enable(TimeStep) decimos que queremos activar el componente y que utilice la frecuencia de actualización en el tiempo TimeStep.
- Obtener un valor del componente(si lo hay): 'componente'.getValue/s()[posición]



Armar nuestro controlador

Como hemos visto, las instrucciones se basan en un patrón hasta cierto punto. El realizar nuestro controlador implica imaginar y crear los pasos que nosotros deseamos que realice el robot. No existen limitaciones en esta tarea de creación, ustedes pueden crear lo que imaginen y les parezca adecuado sin verse obligados a limitarse, **no existe nada imposible para ustedes.**