

PRÁCTICA III

Pablo Chantada Saborido & José Romero Conde

Introducción

Como configuración de la aplicación de Robobo utilizamos "Random Behaviour in Worlds" con "Horizontal Flip Front Camera" desactivado. Para evitar la repetición de código, implementamos una clase base `RoboboController` que implementa la funcionalidad común, y clases derivadas para cada tipo de interfaz: `TextRoboboController`, `VoiceRoboboController`, `ChatGPTRoboboController` y `VectorRoboboController`.

Cada controlador implementa su propio método `get_command()` para obtener instrucciones del usuario, mientras que comparten la lógica de movimiento, emociones y efectos.

Ejercicio 1: Movimiento con Teclado

Este ejercicio implementa un controlador básico del robot Robobo mediante teclado utilizando la biblioteca `pynput`.

Control: W (adelante), S (atrás), A (izquierda), D (derecha), Q (detener y salir).

El robot utiliza una velocidad constante (`SPEED=10`) y detiene los motores al soltar cualquier tecla. Los movimientos de izquierda y derecha giran el robot de forma estacionaria.

Ejercicio 2: Movimiento con Cuadro de Texto

Este ejercicio permite controlar el robot Robobo mediante comandos de texto introducidos en una interfaz gráfica con cuadros de diálogo.

Movimientos básicos: forward, back, left, right.

Movimientos compuestos: forward-left, forward-right, back-left, back-right.

Admite parámetros adicionales:

- speed X: Establece velocidad a X
- time Y: Establece duración a Y segundos

La implementación se basa en la clase `TextRoboboController` que hereda de la clase base `RoboboController`, modificando únicamente la obtención de los comandos mediante el diálogo anteriormente mencionado.

Ejercicio 2b: Movimiento con STT

Apartado opcional del ejercicio 2 que utiliza reconocimiento de voz en español y añade control de emociones y LEDs.

Movimientos: delante, atrás, izquierda, derecha.

Emociones: feliz, triste, enfadado.

Colores LED: rojo, verde, azul.

Utilizando la biblioteca `speech_recognition` con reconocimiento Google (`language="es"`). Cambiamos el idioma de los comandos para facilitar la comprobación de su funcionalidad, el tiempo de recepción de comandos debe ser cambiado en el propio código; inicialmente está puesto a 4,5 segundos.

Ejercicio 3: Definición de prompt para ChatGPT

Este ejercicio consiste en la definición de un prompt específico para ChatGPT que permite extraer comandos de movimiento para el robot a partir de frases en lenguaje natural. A continuación se muestra el contenido del prompt que utilizamos para el entrenamiento:

```

1 Necesito que extraigas comandos de movimiento para un robot a partir de
   frases en lenguaje natural.
2
3 Los comandos permitidos son:
4 - forward: para mover hacia adelante
5 - back: para mover hacia atras
6 - left: para girar a la izquierda
7 - right: para girar a la derecha
8 - stop: para detener el movimiento
9
10 Tambien puedes combinar los comandos de direccion para movimientos
   compuestos como:
11 - forward-left: para avanzar girando a la izquierda
12 - forward-right: para avanzar girando a la derecha
13 - back-left: para retroceder girando a la izquierda
14 - back-right: para retroceder girando a la derecha
15
16 Ademias, puedes incluir parametros:
17 - speed X: donde X es un numero entero que indica la velocidad
18 - time Y: donde Y es un numero que indica los segundos de movimiento
19
20 Ejemplos:
21 - "Avanza un poco": forward
22 - "Retrocede a velocidad 30": back speed 30
23 - "Gira a la derecha durante 3 segundos": right time 3
24 - "Avanza girando a la izquierda a velocidad 25": forward-left speed 25
25
26 Responde UNICAMENTE con el comando extraido, sin explicaciones ni formato
   adicional.
```

La ejecución del entrenamiento se puede comprobar al final del informe.

Ejercicio 4: Movimiento con ChatGPT

Este ejercicio integra ChatGPT para interpretar comandos en lenguaje natural y convertirlos en instrucciones para el robot.

El usuario introduce una frase (ej: "Avanza un poco"), ChatGPT la procesa y devuelve un comando (indicado en el ejercicio 3, ej: "forward"), que el controlador ejecuta. Utilizamos temperatura 0 en este caso, ya que queremos únicamente los comandos establecidos.

Ejercicio 5: Movimiento con vector de ChatGPT

Este ejercicio, a diferencia del anterior, genera un vector de control completo para el robot.

Vector de control: [vx, vy, emoción, sonido, texto]

Donde vx, vy son las velocidades de las ruedas (-100 a 100), emoción y sonido pueden ser "happy", "sad" o "angry", y texto es un mensaje que el robot emite.

El controlador ejecuta todos los aspectos simultáneamente y detiene el robot después de 3 segundos. Fue necesario adaptar el entrenamiento del ejercicio 3 usando el siguiente texto:

```

1 Necesito que generes comandos de control para un robot Robobo a partir de
   frases en lenguaje natural.
2
3 Tu respuesta DEBE ser un vector de 5 elementos con el siguiente formato
   exacto: [vx, vy, emocion, sonido, texto]
4
5 Donde:
6 - vx y vy son valores numericos entre -100 y 100 que representan las
   velocidades de las ruedas derecha e izquierda, usa 20 como valor por
   defecto
7 - emocion es un string que debe ser uno de los siguientes: "happy", "sad", "
   angry"
8 - sonido es un string que debe ser uno de los siguientes: "happy", "sad", "
   angry"
9 - texto es un string con el mensaje que el robot debe decir
10
11 Ejemplos de comandos y sus respuestas:
12 - "Avanza rapido con cara feliz": [50, 50, "happy", "happy", "Avanzando a
   toda velocidad!"]
13 - "Gira a la derecha lentamente con cara triste": [-20, 20, "sad", "sad", "
   Girando a la derecha..."]
14
15 Para los movimientos, considera:
16 - Avanzar: valores positivos iguales en vx y vy
17 - Retroceder: valores negativos iguales en vx y vy
18 - Girar a la derecha: vx positivo, vy negativo
19 - Girar a la izquierda: vx negativo, vy positivo
20 - Velocidad rapida: valores cercanos a 100
21 - Velocidad lenta: valores cercanos a 20
22
23 Responde UNICAMENTE con el vector, sin explicaciones ni texto adicional.
```

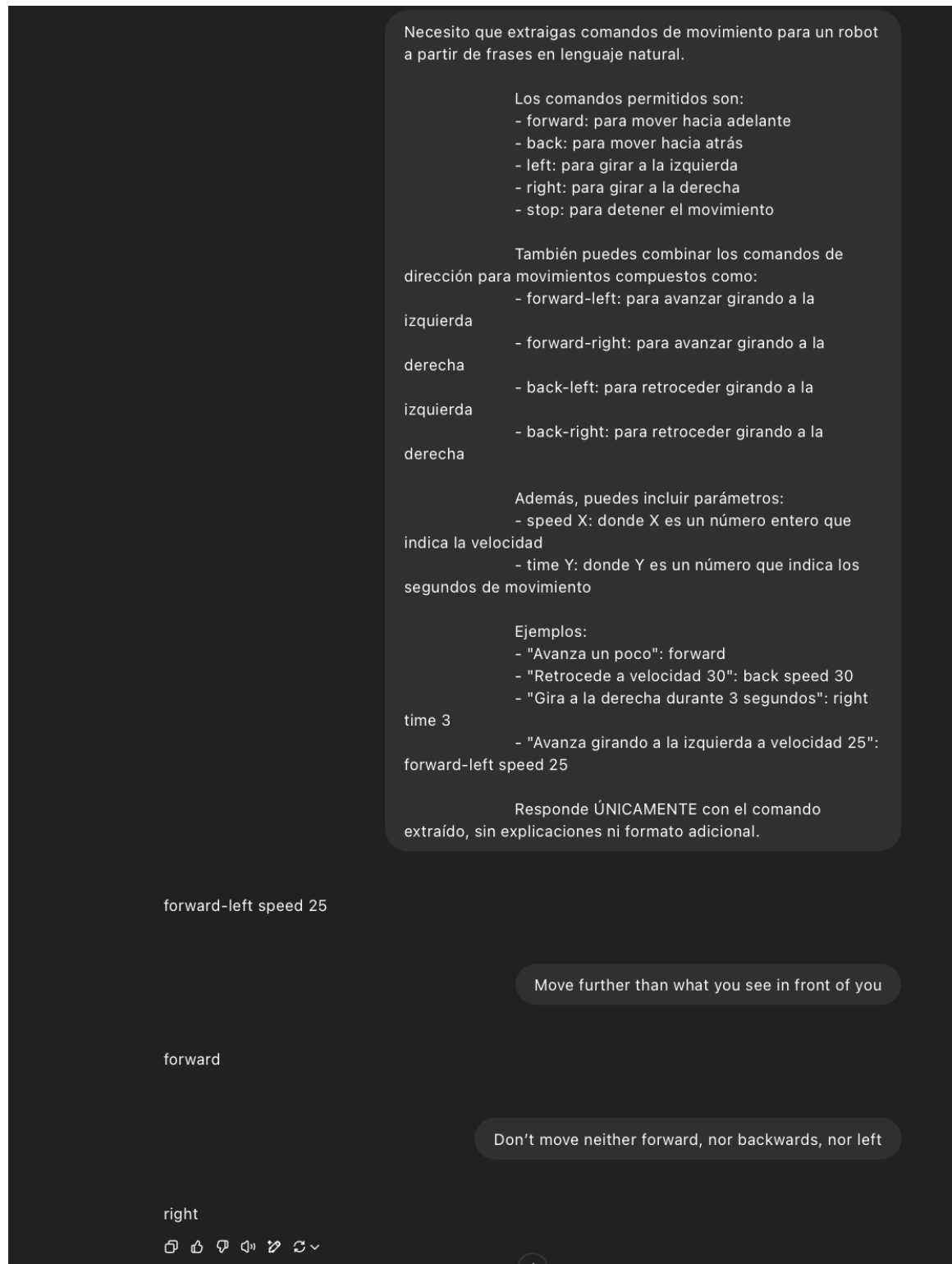


Figure 1: Entrenamiento de ChatGPT para movimiento de Robobo