

PRÁCTICA II

Pablo Chantada Saborido & José Romero Conde

Resolución de los ejercicios

En ambos ejercicios, el control del robot ocurre en la llamada:

```
robobo.whenANewColorBlobIsDetected(blobDetectedCallback)
```

y por tanto en la correspondiente función `blobDetectedCallback`, esta es igual para ambos casos. Este callback a su vez llama a otras dos funciones para manejar el comportamiento del robot: `centerToAColor(blob)` y `moveToAColor(blob)`. Las cuales están diferentemente implementadas para cada ejercicio.

Ejercicio 1 - *Control Proporcional*

Función `centerToAColor(blob)`

Dentro de esta función, el robot centra el blob según el error de los sensores con el centro (*se considera centro como el valor medio del rango $[0,100]$*). Añadimos además un leve margen de error para evitar bucles infinitos por precisión ¹. La velocidad a la que gira para centrarse es fija, independiente del error.

Función `moveToAColor(blob)`

Se implementa el control proporcional. Por tanto, tenemos un coeficiente K_{Pav} y una variable P que es igual al sensor IR frontal. Con esto, calculamos la corrección ε_t ($:= P \times K_{Pav}$) de forma que

$$v_t := v_{t-1} - \varepsilon_t$$

siendo v_t la velocidad del robot en el instante t .

Ejercicio 2 - *Control PID*

En el anterior caso cuando P (*la variable que carga el error actual*) era menor que un cierto valor, la ejecución se paraba; ahora se llama a otra subrutina `blob_is_close(speed, distance)` que se encarga de mover el blob si estamos seguros de estar cerca; por último, se detiene la ejecución.

Función `centerToAColor(blob)`

Ahora, la velocidad de giro para centrarse sí depende del error, siguiendo un esquema PID. Para ello, definimos P_i como el error en el eje x (`abs(blob.posx - 50)`). Adicionalmente ahora cargamos con dos variables más (y sus coeficientes), que se definen de la siguiente forma

$$I_t = \sum_{i \in [0,t]} P_i$$

$$D_t = P_t - P_{t-1}$$

con esto, definimos la nueva corrección, que usamos para el movimiento del robot para que el blob esté constantemente centrado:

$$\varepsilon_t := P_t \times K_p + I_t \times K_I + D_t \times K_D$$

¹El robot se queda moviéndose de derecha a izquierda por no encontrar el valor exacto.

Función `moveToAColor(blob)`

Seguimos la misma estrategia de PID descrita, pero ahora siendo P como se definió en el ejercicio 1.

Ajuste de controladores

Los valores de los controladores han sido ajustados manualmente, viendo el comportamiento en el simulador. Únicamente se han usado los mapas indicados en la guía; en otros escenarios podría ser necesario un ajuste a estos.