

Simulación de control de un robot

Daniel Fernández Pérez
Miguel García-Mauriño Taboada

4 de marzo de 2019

Índice

1. Seguimiento de la línea	2
1.1. Consigna de control para el seguimiento de la línea: PID	2
1.2. Estimación de las constantes de los PID	2
1.3. Otras consideraciones al seguimiento de la línea	2
1.4. Límites del sistema	3
1.5. Pruebas realizadas para el seguimiento de la línea	4
2. Esquive de obstáculos	8
2.1. Seguir una pared	8
2.2. Rodear una esquina	8

1. Seguimiento de la línea

1.1. Consigna de control para el seguimiento de la línea: PID

Para seguir la línea, hemos programado un controlador PID (`./pid.py`) que utilizamos para controlar tanto la velocidad como la rotación del robot. La entrada de `.error` para la rotación es la diferencia entre la posición de la línea y el centro del robot, mientras que para la velocidad la entrada es la rotación. Queremos que la velocidad sea mayor cuanto menor sea la rotación, por lo que el PID que la controla debe ser inverso. Así, la consigna de control del robot se calcularía de la siguiente forma:

$$r = K_{pr} \times \varepsilon + K_{ir} \times \int_t \varepsilon dt + K_{dr} \times \frac{d\varepsilon}{dt} \quad (1)$$

$$v = K_{pv} \times r + K_{iv} \times \int_t r dt + K_{dv} \times \frac{dr}{dt} \quad (2)$$

1.2. Estimación de las constantes de los PID

Para estimar las constantes de los PID seguimos el procedimiento Ziegler-Nichols simplificado, como se ha descrito en clase. Para ello, hacemos que el cerebro del robot escriba en un fichero el error que comete con respecto a la línea en cada momento, y representamos el contenido de dicho fichero como una gráfica (`./log.py`), de manera que podemos localizar las oscilaciones de forma sencilla.

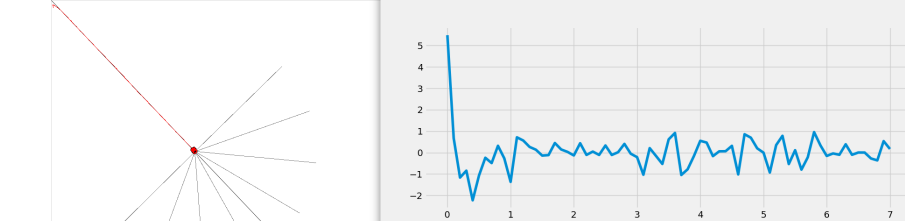


Figura 1: Ejemplo del robot siguiendo una línea recta, con su gráfica de error adjunta

1.3. Otras consideraciones al seguimiento de la línea

Otro de los cambios que realizamos fue la reacción del robot cuando pierde la línea. En este caso, en lugar de parar el robot, distinguimos entre dos situaciones:

1. Si el robot acaba de iniciarse, es decir, nunca ha encontrado una línea, comienza a describir una espiral hasta que la encuentra.
2. Si el robot ya se encontraba en una línea, entonces sabemos que la línea no puede estar lejos, por lo que se para y rota hasta que la encuentra.

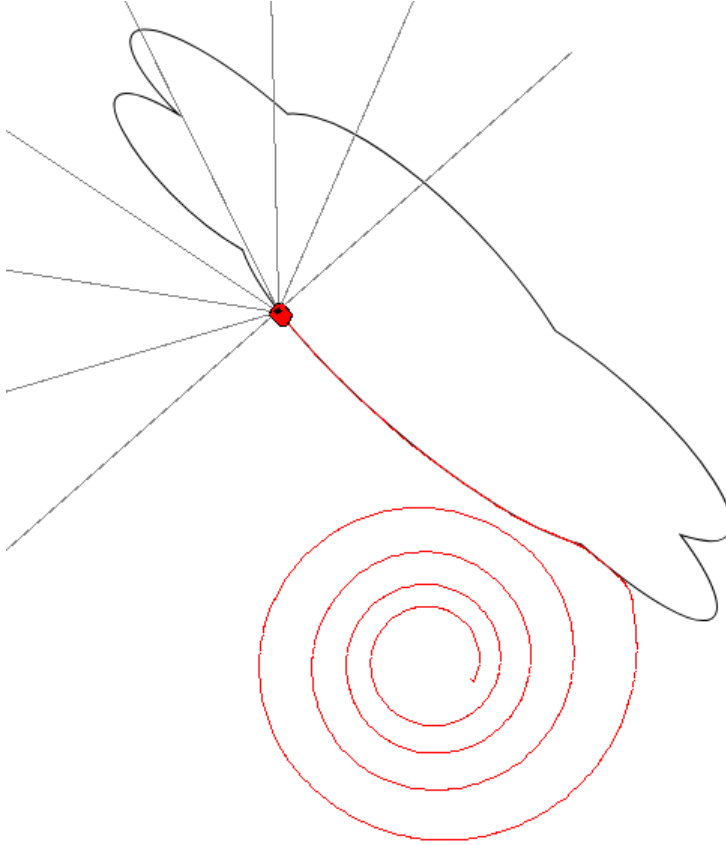


Figura 2: Caso 1: El robot describe una espiral hasta que encuentra la línea

1.4. Límites del sistema

El control no funciona todas las veces cuando se encuentra una curva muy cerrada, puesto que en algunos casos detecta las dos líneas como una sola y calcula el centro de esta línea virtual. Además, si encuentra la línea de forma perpendicular a su trayectoria, necesita varios intentos, como se muestra en la siguiente imagen

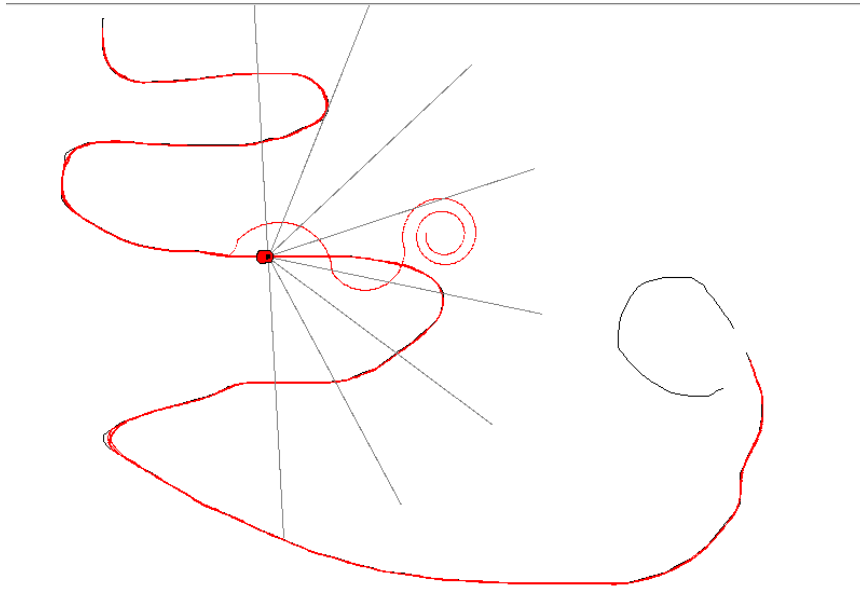


Figura 3: Error para seguir la línea cuando se encuentra de forma perpendicular

1.5. Pruebas realizadas para el seguimiento de la línea

Además de los ya mostrados, a continuación se muestran los recorridos del robot en diferentes mapas.

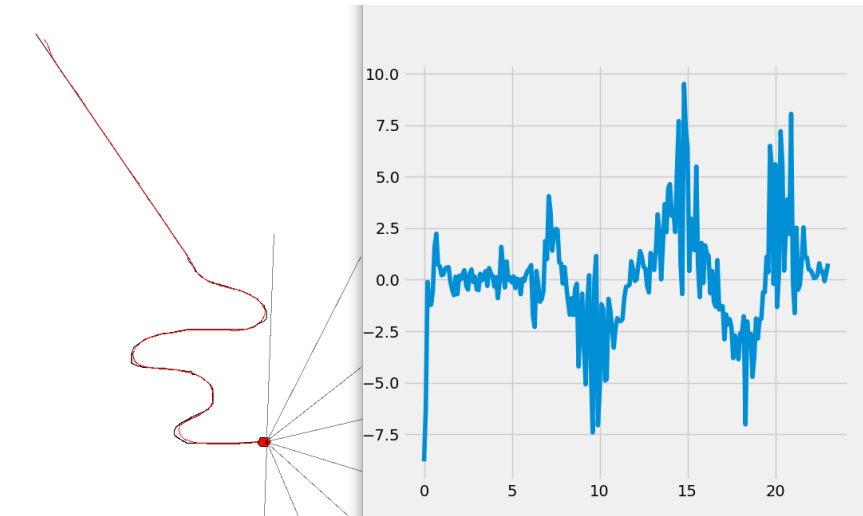


Figura 4: Recorrido por el circuito inicial, con gráfica de error

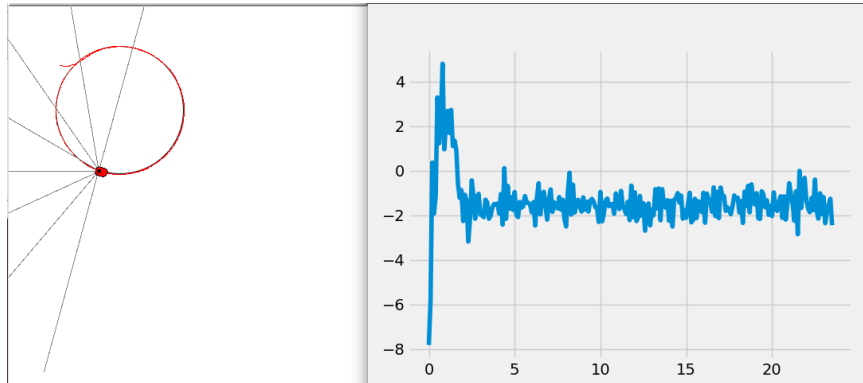


Figura 5: Recorrido por una circunferencia, con gráfica de error

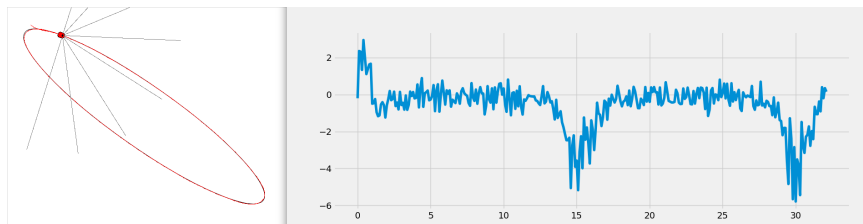


Figura 6: Recorrido por una elipse no muy excéntrica, con gráfica de error

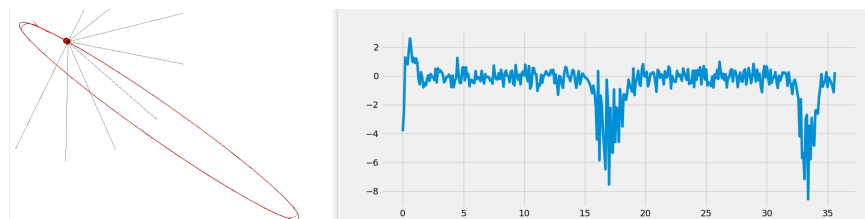


Figura 7: Recorrido por una elipse de mayor excentricidad, con gráfica de error

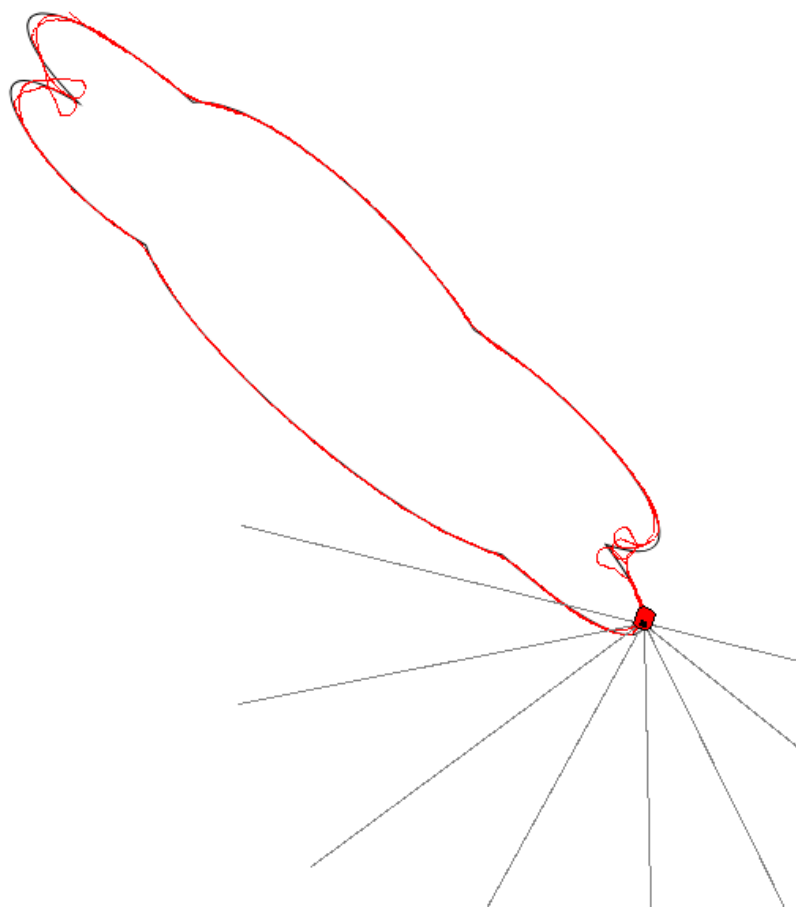


Figura 8: Recorrido por una figura como una nube, con errores en las curvas más cerradas

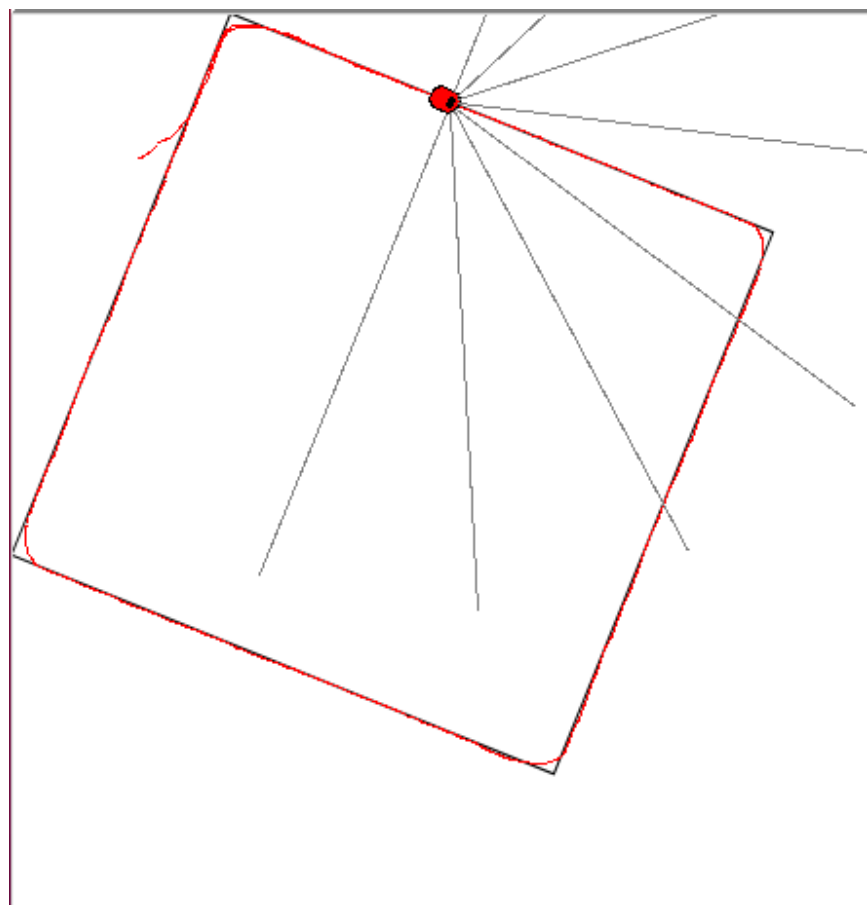


Figura 9: Recorrido por un cuadrado

2. Esquive de obstáculos

Para detectar y esquivar obstáculos utilizamos los sonar del robot. Diferenciamos entre dos estados: seguir una pared y girar una esquina cóncava con respecto al robot.

2.1. Seguir una pared

Para seguir la pared, intenta mantener la distancia con la misma dentro de un rango, de esta manera, si se detecta que la distancia es menor que un umbral, se gira para alejarse y viceversa cuando se detecta una distancia mayor. A partir de la definición, se observa que este mismo estado sirve para girar las esquinas convexas con respecto al robot, como las que se encuentra cuando rodea una caja rectangular. Cuando se encuentra una pared de frente suponemos que se haya en una esquina cóncava, como las que se puede encontrar cuando sigue las paredes por dentro de una habitación. En este caso, pasa al siguiente estado

2.2. Rodear una esquina

Tal y como está hecho el programa, el robot gira siempre en el mismo sentido, por lo que siempre podemos suponer que tiene la pared al mismo lado y, por lo tanto, siempre que se encuentra una pared de frente gira en sentido contrario hasta que detecta que la distancia frontal con la pared es suficientemente grande.