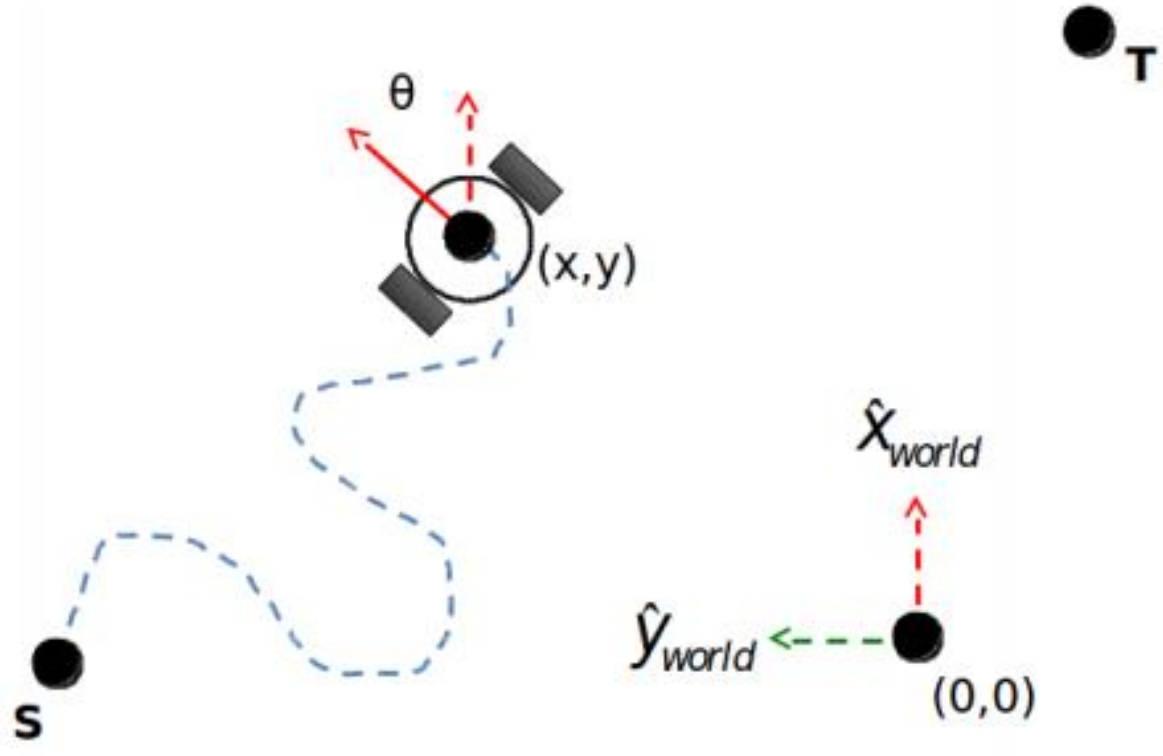
PRÁCTICA 4: LOCALIZACIÓN GEOMETRICA



Melania Prieto Martín 100292147

INTRODUCCIÓN:

El objetivo de esta práctica es la comprensión de los métodos para conocer la localización de un robot, en especial la odometría, aunque también se menciona el uso del GPS. A través de la realización de esta práctica se nos presenta la oportunidad de experimentar las diferentes posibilidades de los distintos métodos. Esto es, poder usar la odometría basándonos directamente en las ecuaciones que definen el movimiento del robot, o por ejemplo, mediante el uso de codificadores rotacionales, incluidos en las ruedas del robot Pioneer II.

En la siguiente imagen podemos ver la formula general para el cálculo de la posición de forma general, usando la odometría.

$$p = f(x, y, \theta, \Delta s_r, \Delta s_l) = \begin{bmatrix} x \\ y \\ \theta \end{bmatrix} + \begin{bmatrix} \frac{\Delta s_r + \Delta s_r}{2} \cdot cos(\theta + \frac{\Delta s_r - \Delta s_l}{2b}) \\ \frac{\Delta s_r + \Delta s_r}{2} \cdot sin(\theta + \frac{\Delta s_r - \Delta s_l}{2b}) \\ \frac{\Delta s_r - \Delta s_l}{b} \end{bmatrix}$$

En el caso de un movimiento recto, en el que la variación del ángulo es despreciable:

$$p' = \begin{bmatrix} x \\ z \\ \theta \end{bmatrix} + \begin{bmatrix} \left(\frac{\Delta s_r + \Delta s_l}{2}\right)\cos(\theta) \\ \left(\frac{\Delta s_r + \Delta s_l}{2}\right)\sin(\theta) \\ 0 \end{bmatrix}$$

En primer lugar, tenemos que programar un controlador qué, mediante odometría, nos permita hacer que el robot se desplace desde el punto inicial del mundo scenario00.wbt, hasta el final del mismo, cruzando la línea amarilla. En segundo lugar, se nos pide un controlador que, igual que en el anterior caso, controle al robot hasta cruzar el mundo, pero esta vez, esquivando un objeto colocado en el mundo scenario_odometry.wbt.

RESOLUCIÓN:

Para la realización de esta práctica, me he basado en el uso de la odometría, combinando la fórmula de la posición final, con el uso de codificadores. Para el primer controlador he empleado la formula simplificada para el movimiento en un solo eje para la variable x, mientras que para la variable z, he empleado los codificadores de las ruedas del robot. He hecho esto porque, cómo el movimiento es prácticamente recto (hay ciertas variaciones debido al uso de la brújula), y en el sentido del eje z, el incremento del valor que devuelven los codificadores se produce casi en su totalidad en este eje. Por

otra parte, para el desplazamiento en x he querido demostrar que también es posible el uso de la fórmula expuesta previamente.

No obstante, he realizado algunos cambios, ya que si se usa la fórmula tal como se ve en la figura arriba mostrada, el desplazamiento en el eje x siempre se vería incrementado, independientemente de la dirección de oscilación que el robot tiene debido al bucle que controla el uso de la brújula. Sin embargo, esto no puede ser así, ya que el eje x tiene su cero en el centro del escenario, y por tanto, si el robot lleva a cabo un giro en el sentido negativo de este eje, debe haber un incremento negativo sumado a la posición inicial, y no positivo. Para corregir dicho error, he optado por calcular un ángulo, resultado de la variación entre el ángulo previamente leído por la brújula, y el nuevo ángulo. Además he calculado el seno (en radianes) de este ángulo, y no el coseno, ya que éste resulta positivo en ángulos negativos menores de 90º, cómo es el caso.

Para el uso de los codificadores, y teniendo en cuenta la información incluida en las guías del programa, he multiplicado el radio de las ruedas del robot, por el incremento medido por el codificador (en radianes), y dividido esto entre la resolución del mismo. Además he hecho la media entre los valores izquierdo y derecho, para aumentar la precisión.

Dicho esto, he obtenido valores cercanos a los mostrados por el propio programa. Cabe destacar que aun así, el valor obtenido mediante el uso de encoders tiene menos error. Por este motivo, en el segundo controlador, me he centrado en este método para calcular la posición del robot.

En el segundo apartado, he programado el controlador con condiciones de distancias y estados para esquivar el objeto. Al cambiar el eje de desplazamiento, he empleado la función setEncoders para reinicializarlos, ya que miden el ángulo recorrido por la rueda (más de una revolución). En el eje en el que no existen variaciones significativas de posición, he asumido un valor constante medio.

CONCLUSIÓN:

Podemos concluir que la odometría por sí misma no es exacta, y conlleva dificultades de control. Con la ayuda de dispositivos cómo los codificadores, la precisión es mayor, aunque el error sigue siendo acumulativo si no disponemos de medidas intermedias para reducir errores. Además, es importante tener en cuenta que cuando es necesario el movimiento en varios eje, los cálculos serían más complejos, y el resultado menos preciso. Para finalizar, este método no resulta práctico si el robot ha de esquivar múltiples objetos.