



EXPERIMENTO 6

MOVIÉNDOSE A UN PUNTO

PROFESOR: ATOANY FIERRO

UNIDAD DE FORMACIÓN: MR3001B – DISEÑO Y DESARROLLO DE ROBOTS

EXPERIMENTO 6

MOVIÉNDOSE A UN PUNTO

OBJETIVO

Simular la navegación de un robot móvil desde un punto A hasta un punto B utilizando el método de dead reckoning en Simulink. Los estudiantes deberán calcular la distancia recorrida a partir de los datos de los encoders de las ruedas, implementar la simulación en MATLAB/Simulink y utilizar Stateflow para controlar el movimiento del robot, deteniéndolo cuando haya alcanzado una distancia predefinida. Esta actividad

permitirá comprender la estimación de posición basada en sensores, así como la importancia de la integración de señales en la navegación autónoma.

INTRODUCCIÓN

En la navegación de robots móviles, es fundamental estimar con precisión su posición y trayectoria sin depender de señales externas como GPS. Uno de los métodos más utilizados para este propósito es dead reckoning, una técnica que calcula la ubicación actual del robot a partir de una posición inicial conocida, utilizando la dirección y distancia recorrida.

El dead reckoning se basa en la información proporcionada por los encoders de las ruedas, los cuales registran la cantidad de vueltas realizadas por cada rueda y permiten calcular la distancia total recorrida por el vehículo. Sin embargo, debido a que este método acumula errores con el tiempo, es esencial implementar estrategias para mitigar su impacto en la precisión de la navegación.

En esta actividad, los estudiantes desarrollarán una simulación en MATLAB/Simulink para modelar el desplazamiento de un robot móvil utilizando dead reckoning. Mediante el uso de bloques de Simulink, representarán el cálculo de la distancia recorrida a partir de los datos de los encoders y controlarán la detención del robot una vez alcanzada una distancia predefinida, aplicando Stateflow para gestionar los estados del movimiento. Con esta práctica, los estudiantes comprenderán los principios fundamentales de la estimación de posición en robótica móvil y su implementación en entornos de simulación.

MATERIAL Y EQUIPO

- Computadora con capacidad suficiente para ejecutar MATLAB/Simulink sin problemas
- MATLAB y Simulink instalados
- Toolbox simulink (incluyendo Stateflow para la gestión de estados)
- Acceso a internet para consulta de documentación o resolución de duda

INSTRUCCIONES

Dead reckoning se define como una manera de calcular la posición de vehículo usando solamente información sobre la dirección y distancia que se ha desplazado desde un punto conocido, como se muestra en la Figura 11.



Fig. 11. *Dead reckoning* de un robot móvil

Para poder determinar la posición del vehículo es necesario el uso de los encoders, los cuales son dispositivos que nos ayudan a medir las vueltas que da una rueda. Por ejemplo, en la Figura 12 aparece una rueda un encoder. En este caso, el encoder cuenta con 9 dientes los cuales son detectados por sensores ópticos. Cuando el sensor detecta 9 veces un diente diferente, quiere decir que se ha completado una vuelta completa.



Fig. 12. Rueda (izquierda) y encoder (derecha) La rueda gira 360° para dar una vuelta entera, mientras que el encoder realiza 9 detecciones de sus dientes cuando completa una vuelta completa.

Luego entonces, tenemos que:

$$1 \text{ vuelta} = 9 \text{ detecciones de diente} \quad (1)$$

$$\text{Número de vueltas} = \frac{\text{Número total de detecciones de diente}}{\text{Número de dientes por vuelta}} \quad (2)$$

Es decir, si nuestro encoder de 9 dientes lo detecta 23 veces, entonces el número de vueltas que ha dado la rueda es:

$$\text{Número de vueltas} = \frac{23}{9} = 2.5$$

Sabemos que cuando las ruedas dan una vuelta, el vehículo se ha desplazado una distancia equivalente a la circunferencia de la rueda, por lo que:

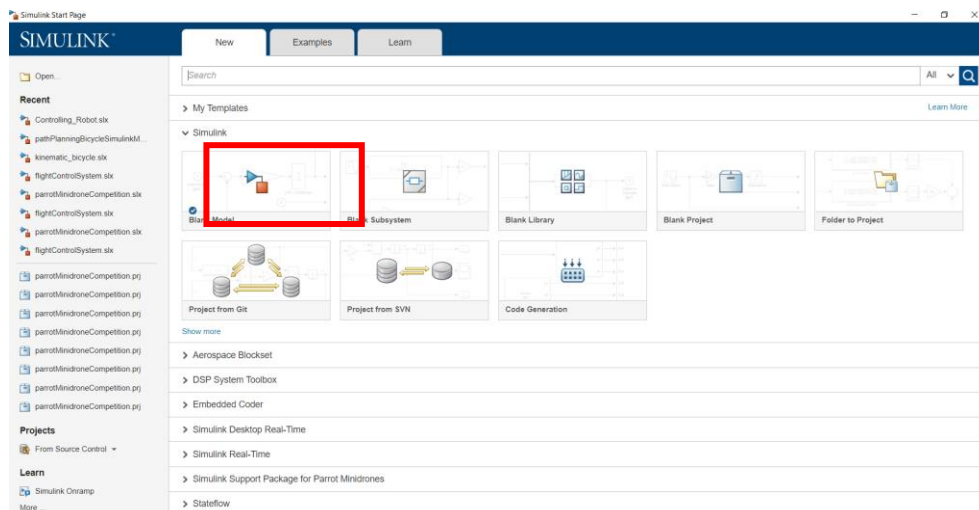
$$1 \text{ vuelta} = 2\pi R \quad (3)$$

Donde R es el radio de la rueda. La distancia total recorrida del vehículo se calcula de la siguiente manera:

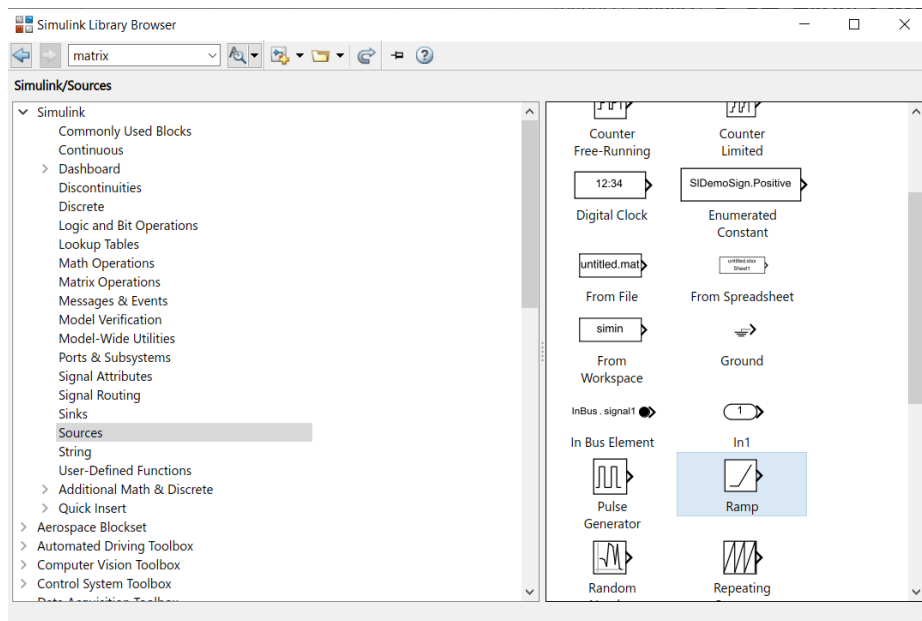
$$\text{Distancia recorrida} = k2\pi R \quad (4)$$

Donde k es el número de vueltas que da la rueda la cual se calcula con la ecuación 30.

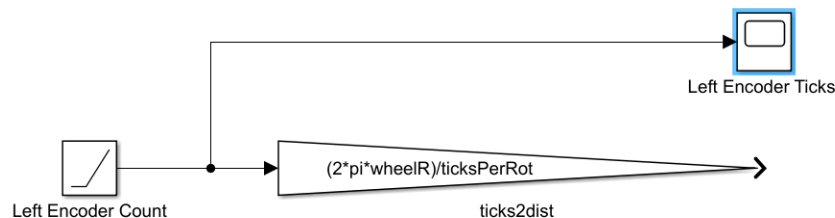
El primer paso para simular es abrir simulink introduciendo en la ventana de comandos de Matlab el comando simulink y presionando enter. De la ventana se simulink, crear un archivo nuevo en blanco.



El siguiente paso es agregar los bloques para realizar la simulación de un robot móvil. Para ello dar clic en Library Browser y seleccionar la categoría sources para seleccionar ramp. Clic en ramp y arrastrar el bloque hacia el espacio de trabajo.



Ir ahora a Math y colocar un bloque gain en el área de trabajo. Colocar ahora un bloque scope el cual se encuentra en la categoría Commonly Used Blocks. El área de trabajo se deberá ver como se muestra a continuación:



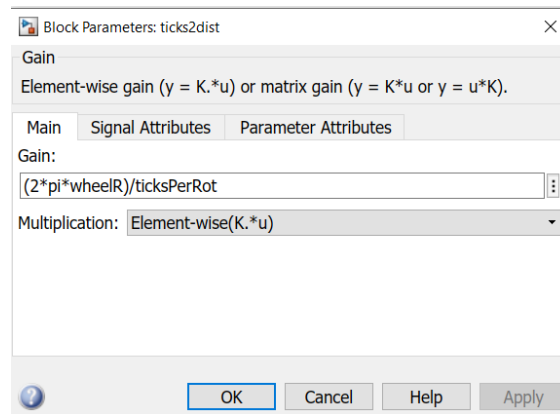
Dar doble clic en el bloque gain para introducir la siguiente ecuación:

$$\frac{2\pi R \times wheelR}{ticksPerRot} \quad (5)$$

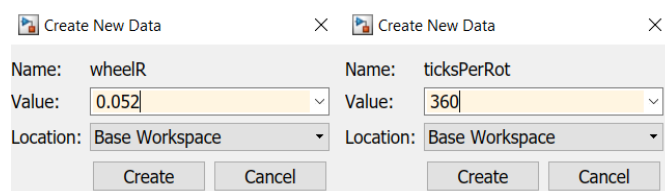
En donde $2\pi R$ es la circunferencia de la rueda, $wheelR$ es el radio, y $ticksPerRot$ es el número de dientes que tiene el encoder. Para introducir la ecuación 33 en código, debemos de hacer lo siguiente:

$$2 * \pi * wheelR / ticksPerRot$$

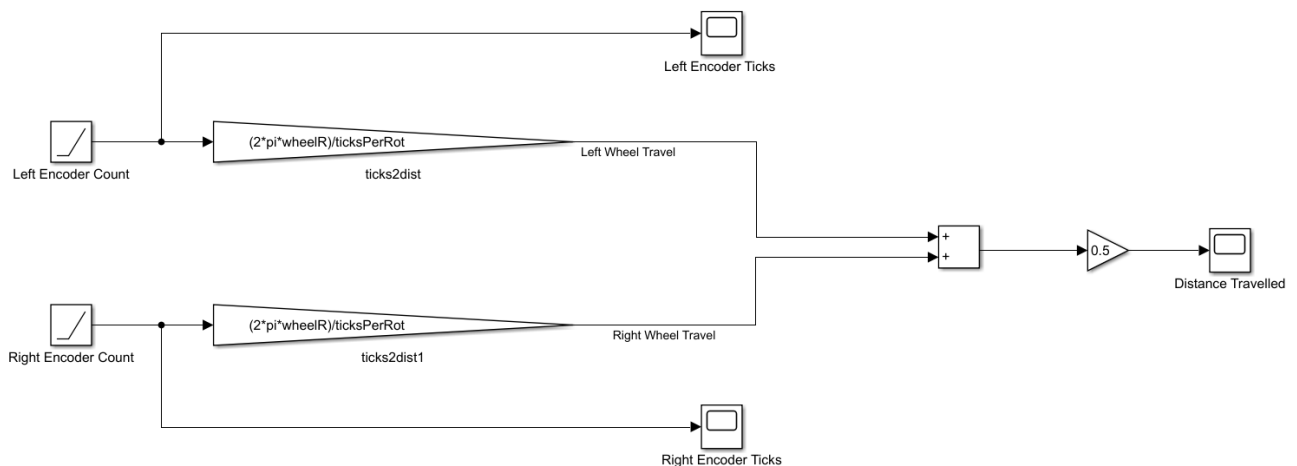
Una vez que introduzcamos la ecuación en el bloque gain se verá como sigue:



A continuación, daremos clic en los tres puntos que se encuentran a la derecha de la ecuación que introdujimos y crearemos las variables correspondientes. Para la variable wheelR le damos un valor de 0.052 y la declaramos en Base Workspace. Para la variable ticksPerRot, la inicializamos con un valor de 360 en el Base Workspace, como se muestra a continuación:

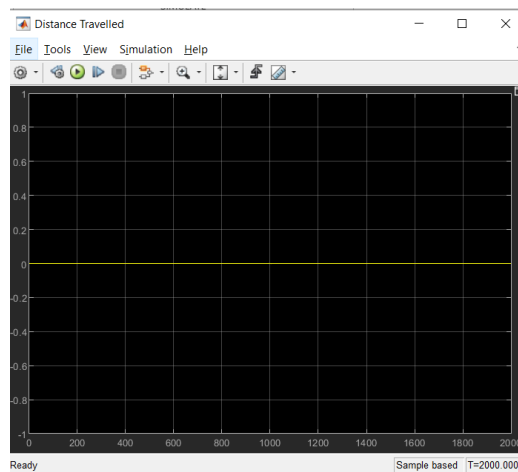


Copiar los bloques una vez para tener controles en la rueda izquierda y la derecha. El espacio de trabajo de deberá ver como sigue:

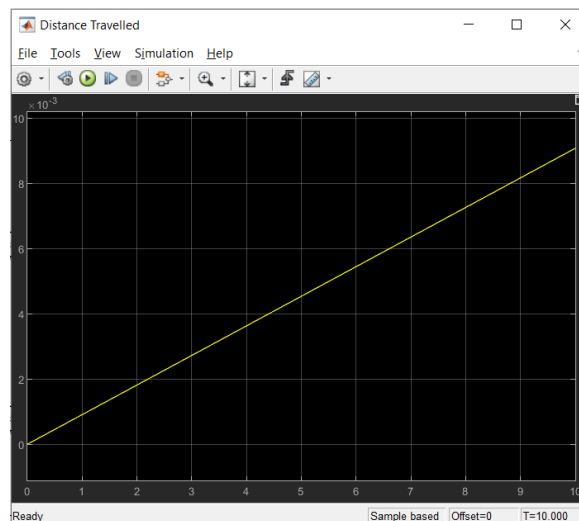


Como se puede observar, existe un bloque add el cual se encuentra en la categoría Math, el cual nos va a permitir sumar las distancias que recorren las ruedas izquierda y derecha. Como sabemos, cuando el vehículo gira, una rueda recorre más distancia que la otra, por lo que la distancia total recorrida es el promedio de las distancias recorridas por cada rueda, es por ello que al final, la suma de las distancias se multiplica por una ganancia de 0.5 o, mejor dicho, se divide entre dos.

Una primera simulación es la de colocar la rampa Left Encoder Count con una pendiente de 1 y la Right Encoder Count como -1. Damos clic en simular y abrimos el scope Distance Travelled. La salida se verá como sigue:

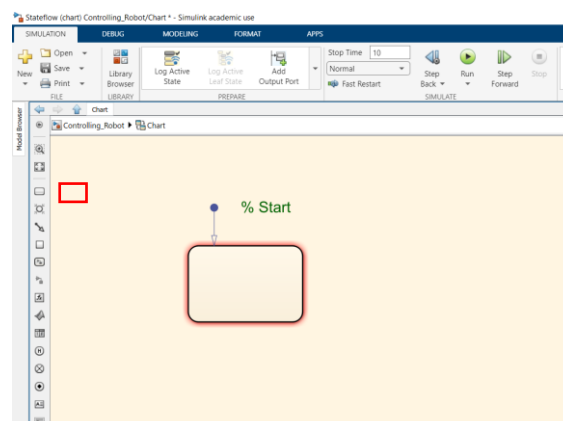


La distancia recorrida es cero, ya que las ruedas giran en sentidos opuestos, por lo que el vehículo solo gira sobre su propio eje. Modificar la pendiente Right Encoder Count con un valor de 1 y volver a simular. El resultado se verá de la siguiente manera:

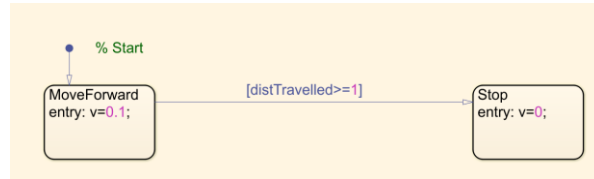


En este caso sí tenemos un desplazamiento ya que las dos ruedas giran en el mismo sentido.

Ahora haremos que el robot se detenga una vez que se ha desplazado un metro. Para ello, haremos uso de State Flow de simulink. Daremos clic nuevamente en la librería y colocaremos un bloque que se llama Chart que se encuentra en la categoría Stateflow. Dar doble clic en el bloque Chart y se nos abrirá una nueva ventana, en donde colocaremos un bloque llamado State, como se muestra a continuación:

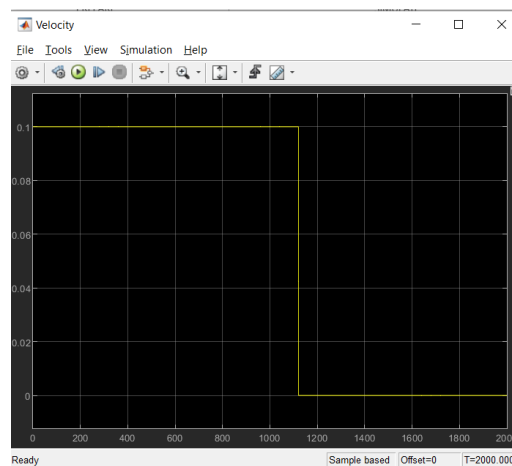
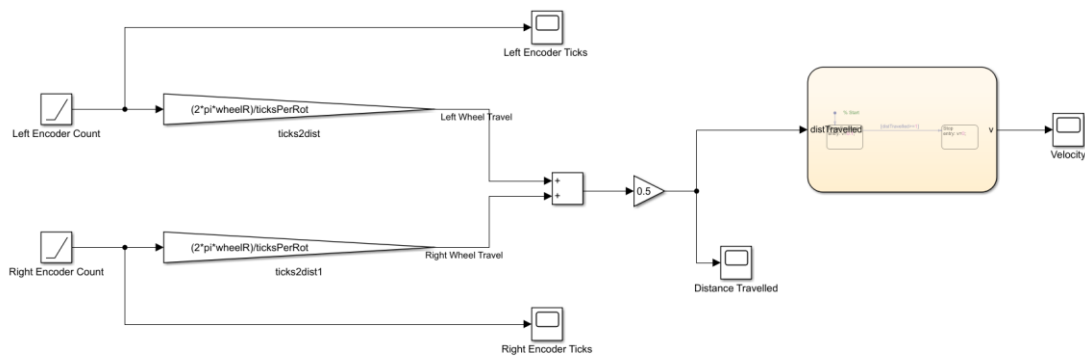


Colocar dos bloques como sigue:



De esta manera estamos diciendo que el robot pasará del estado MoveForward al estado Stop solamente si la distancia recorrida (distTravelled) es mayor o igual a 1. Matlab no reconoce la variable v y tampoco distTravelled, por lo que iremos al menú Modeling y posteriormente en Symbols Pane. Del cuadro de diálogo que se abre a la derecha, dar clic en el ícono rojo de cada variable y luego dar clic en Fix.

El bloque quedará conectado como sigue. Modificaremos el tiempo de simulación a 2000 y le daremos clic en simular. La salida del scope Velocity nos tendría que mostrar que, a partir de 1 metro, la velocidad se hace cero.



ENTREGABLES

- Simulación (archivo en Simulink): 40%
- Video de la simulación (individual) incluyendo una explicación breve: 60%

RÚBRICA DE EVALUACIÓN

La simulación se evaluará con la siguiente rúbrica de evaluación

Criterio	Destacado 95-100	Sólido 85-94	Básico 75-84	Incipiente 0-74
Implementación en Simulink	El modelo está completamente funcional, bien organizado y sin errores.	El modelo funciona correctamente, pero tiene detalles mejorables en la configuración.	El modelo está incompleto o tiene errores en la configuración de los bloques.	No se presenta el archivo o contiene errores graves que impiden su ejecución.
Cálculo de distancia con encoders	La ecuación de distancia se implementa de forma correcta y optimizada.	Se implementa correctamente la ecuación, pero con detalles menores por corregir.	Se implementa parcialmente la ecuación, pero con errores en los cálculos o en la configuración del bloque gain.	No se implementa correctamente la ecuación de distancia recorrida.
Uso de Stateflow para control de movimiento	La implementación en Stateflow es óptima y el robot se detiene en el momento exacto.	Stateflow funciona correctamente, pero con detalles en la transición de estados.	Se implementa Stateflow, pero no detiene el robot correctamente.	No se incluye el bloque Stateflow o no funciona correctamente.
Organización y presentación del modelo	El modelo es claro, bien estructurado y con etiquetas adecuadas en todos los bloques.	El modelo es claro y bien estructurado, con algunas mejoras posibles en la organización.	El modelo es funcional, pero con conexiones desordenadas o etiquetas confusas.	El modelo es desordenado y difícil de interpretar.

El video individual se evaluará con la siguiente rúbrica de evaluación

Criterio	Destacado 95-100	Sólido 85-94	Básico 75-84	Incipiente 0-74
Claridad en la explicación	La explicación es clara, bien organizada y demuestra un dominio profundo del tema.	Se explica correctamente el procedimiento con una estructura clara.	Se explica el procedimiento, pero con omisiones o falta de claridad en algunos puntos.	La explicación es confusa o incompleta, sin describir correctamente el procedimiento.
Demostración de la simulación	La simulación se presenta sin errores y con una excelente explicación de cada paso y resultado.	Se muestra la simulación correctamente y con una explicación adecuada de los resultados.	Se muestra la simulación, pero con errores o falta de detalles clave.	No se muestra la ejecución de la simulación o el video no es funcional.
Análisis de resultados	Se presenta un análisis detallado, reflexionando sobre el comportamiento del robot y posibles mejoras.	Se analizan los resultados correctamente con observaciones adecuadas.	Se menciona el comportamiento del robot, pero sin análisis profundo o con errores.	No se analiza el comportamiento del robot o la información es incorrecta.

Criterio	Destacado 95-100	Sólido 85-94	Básico 75-84	Incipiente 0-74
Presentación y lenguaje	La presentación es profesional, con excelente comunicación y uso adecuado del lenguaje técnico.	La presentación es clara y con buen uso de lenguaje técnico.	La presentación es aceptable, pero con deficiencias en fluidez o lenguaje técnico.	El video es poco profesional, con pausas excesivas, lenguaje inadecuado o desordenado.
Originalidad y autenticidad	La explicación es completamente individual, con un enfoque personal y evidencia de comprensión única.	La explicación es original, con algunas similitudes esperadas en términos técnicos.	La explicación es similar a la de otros estudiantes, con diferencias mínimas.	La explicación es idéntica a la de otros estudiantes o hay evidencia de plagio.