PRIMERA PRÁCTICA
MODELO VEHÍCULO DIFERENCIAL

MIGUEL IAN GARCÍA POZO

GIERM
ESCUELA DE INGENIERÍAS INDUSTRIALES

UNIVERSIDAD DE MÁLAGA

Índice

1.- Simulación del modelo de un vehículo diferencial en Simulink

- Desarrollo:

El esquema final del sistema es el siguiente:

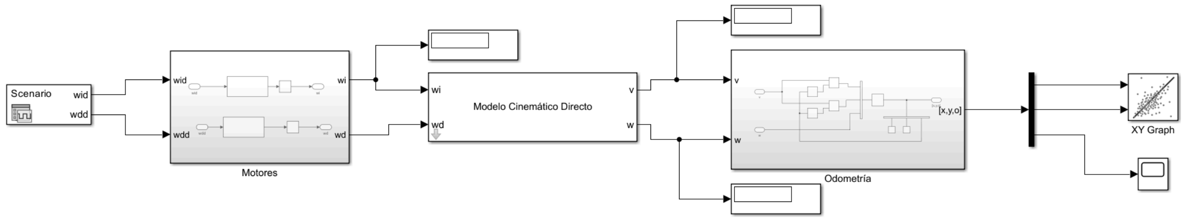


Figura 1.- Esquema Simulink

Entremos en cada parte para verlo con más detenimiento. Primero veamos el generador de señales con el que definimos la trayectoria que queremos que siga nuestro vehículo como podemos ver en la Figura 1.

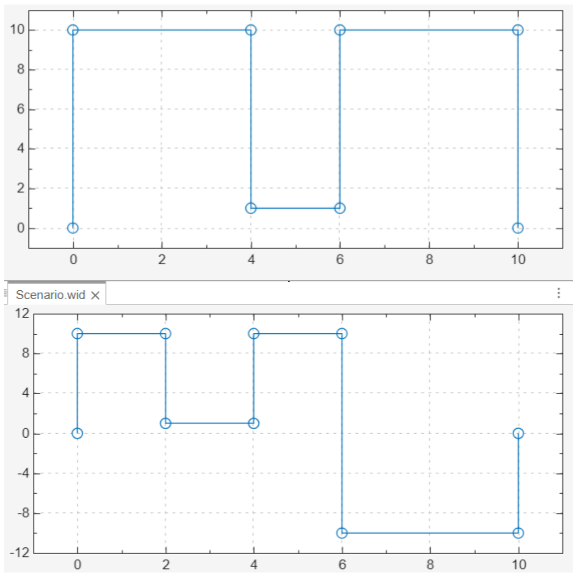


Figura2.- Trayectoria en el generador de señales

Con estas dos señales definimos la rotación de cada una de las ruedas para que el robot siga una trayectoria primero recta, luego un giro a la izquierda, luego uno a la derecha y por último gira con velocidad lineal cero sobre su mismo eje.

A continuación tenemos la función de transferencia del motor de cada una de las ruedas. Para obtenerla debemos hacer varios cálculos:

$\tau_{wi}\cdot w_i' = -w_i + k_{wi}\cdot w_{id}$ $\longrightarrow$ Haciendo la transformada de Laplace obtenemos $\longrightarrow$ $\tau(s\cdot W_i(s) - w_i(0)) + W_i(s) = k_{wi}W_{id}(s)$ $\rightarrow$ Despejando la salida frente a la entrada tenemos: $\frac{W_i(s)}{W_{id}(s)} = \frac{k_{wi}}{\tau s+1}$.

Ya tenemos la función de transferencia en continua, ahora para pasarla a continua debemos hacer lo siguiente:

$G(z) = z[G_{H0} \cdot G(s)] = z[\frac{1-e^{-Ts}}{s} \cdot \frac{k/\tau}{s+1/\tau}] = \frac{(1-z^{-1}) \cdot (1-e^{-k\frac{T}{\tau}})z^{-1}}{(1-z^{-1}) \cdot (1-e^{-k\frac{T}{\tau}}z^{-1})} = \frac{(1-e^{-k\frac{T}{\tau}})z^{-1}}{(1-e^{-k\frac{T}{\tau}}z^{-1})}$ → Sustituyendo por un tiempo de muestreo de 0.025 segundos, una constante de tiempo de 0.12 segundos y una constante k de valor 1, obtenemos la función de transferencia en discreta del sistema:

$\frac{0.1881z^{-1}}{(1-0.8119z^{-1})} = \frac{0.1881}{z - 0.8119}$.

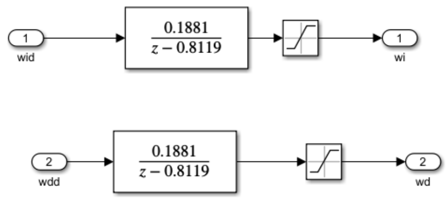


Figura 3.- Modelo de las ruedas

Esta función de transferencia discreta define el comportamiento del motor en un tiempo de muestreo de 0.025 segundos. Aquí también se realiza la saturación de las ruedas en caso de superar su velocidad angular máxima. Después, en la Figura 4 tenemos el "Modelo Cinemático Directo" con el que definimos las velocidades lineales y angulares a partir de las velocidades angulares de las ruedas.

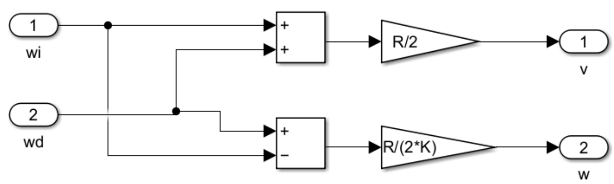


Figura 4.- Modelo Cinemático Directo

Las variables R y K las definimos en la máscara de este subsistema en la Figura 5.

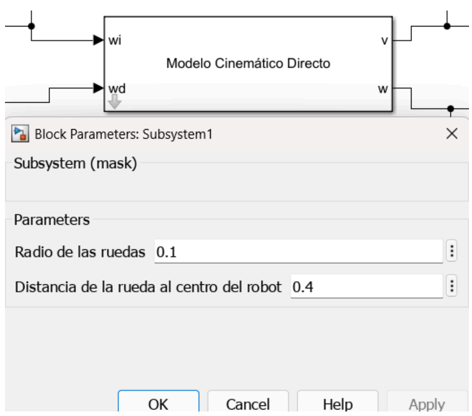


Figura 5.- Variables del radio y distancia de las ruedas

Ahora llegamos a la odometría, donde calculamos, según la distancia que recorremos, la posición y orientación del robot, como podemos ver en la Figura 6.

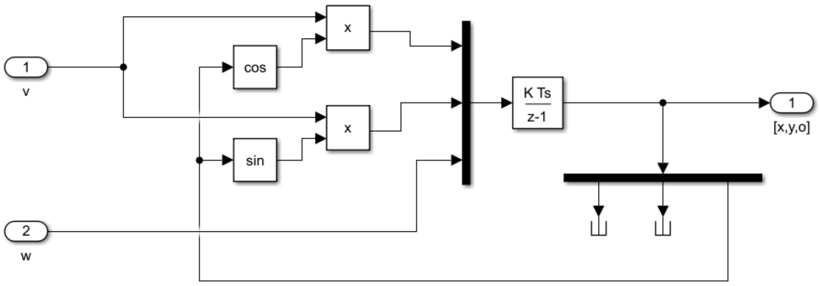
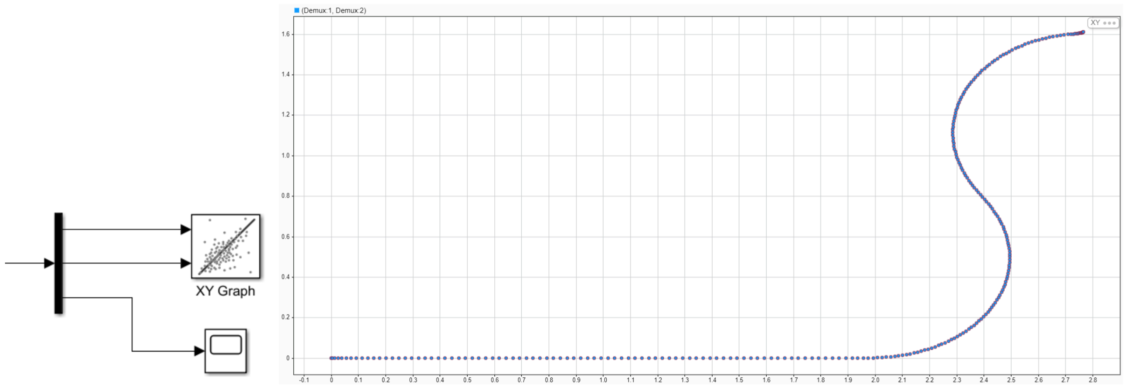


Figura 6.- Odometría

- Simulación:

Habiendo terminado todo el sistema en simulink, podemos sacar ya la posición (x,y) y la representamos para observar los movimientos de nuestro robot en la Figura 8.



Figuras 7 y 8.- Visualización del recorrido del robot

También podemos sacar la orientación en la Figura 9:

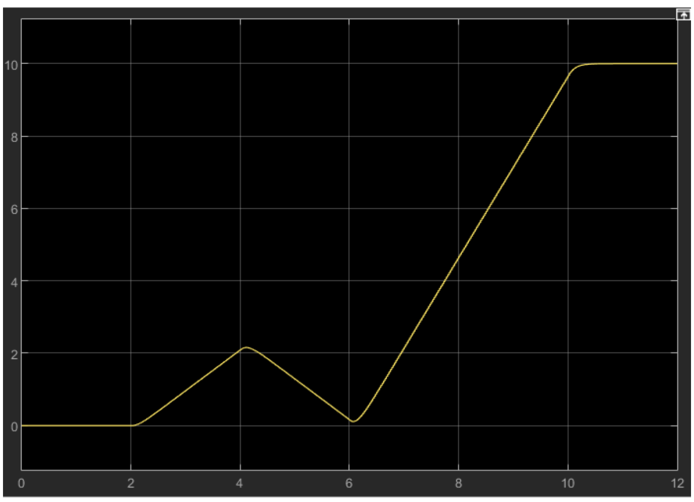


Figura 9.- Gráfica de la orientación

Con la Figura 9 podemos ver como, aunque en la primera gráfica no se pueda observar, el vehículo se queda dando vueltas sin velocidad lineal alguna.

2.- Simulación de un vehículo diferencial en Matlab

- Desarrollo:

Ahora vamos a simular el mismo robot móvil que hemos hecho en Simulink, a diferencia de que lo vamos a hacer en código en Matlab. Primero, inicializamos las variables para realizar los cálculos, y definimos las velocidades angulares deseadas de cada una de las ruedas, como podemos ver en la Figura 10.

```
x = 0;
y = 0;
o=0;
K = 0.8; % m
r = 0.1; % m
tau = 0.12; % s
dt = 0.025; % s
```

```
wid = 0;
wdd = 1;
```

Figura 10.- Variables iniciales

Luego calculamos la función de transferencia de la Planta del sistema, el motor,en tiempo discreto con un tiempo de muestreo de 0.025 segundos y obtenemos las velocidades angulares reales de las ruedas.En este caso, podemos hacer uso del comando "c2d" que nos saca la función de transferencia discreta de la función de transferencia continua de los motores que hemos calculado antes. Esto lo vemos en la Figura 11.

```
G = tf(1/tau,[1 1/tau]);
Gz = c2d(G,dt);
wi = wid*step(Gz,2);
wd = wdd*step(Gz,2);
```

Figura 11.- Función de transferencia

Debemos comprobar que ninguna rueda gire más allá de su límite de velocidad en la Figura 12.

```
if (wi>15)
    wi = 15;
end
if (wi<-15)
    wi = -15;
end
if (wd>15)
    wd = 15;
end
if (wd<-15)
    wd = -15;
end
```

Figura 12.- Código para asegurar una velocidad angular máxima

A continuación, en la Figura 13, calculamos el Modelo Cinemático Directo del robot y obtenemos sus velocidades lineal y angular.

```
v = (wi+wd)*r/2;
w = (wd-wi)*r/K;
```

Figura 13.- Modelo Cinemático Directo

Por último, calculamos los desplazamientos angulares y lineales que se han producido y los utilizamos para calcular la posición y la orientación del robot en el plano. Con el bucle for hacemos la simulación del avance del robot en la Figura 14.

```
deltas = v*dt;
deltao = w*dt;

figure
plot(0,0,"*b")
hold on
for i=1:1500
    o = o + deltao;
    deltax = cos(o).*deltas;
    deltay = sin(o).*deltas;
    x = x + deltax;
    y = y + deltay;
    plot(x,y,"*r")
end
```

Figura 14.- Cálculo de la Odometría

- Simulación:

La simulación la vemos en la Figura 15.

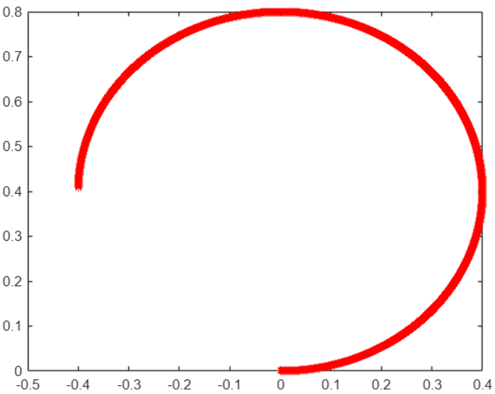


Figura 15.- Simulación del comportamiento del robot

Enlace a Github

https://github.com/MiguelIIan/AmpliacionRobotica/tree/main/Rob%C3%B3tica%20m%C3%B3vil/VehiculoDiferencial