

Actividad 3.1 (Trayectorias en lazo abierto)

En el mundo de la robótica, las trayectorias son muy importantes. Son como la ruta que sigue un robot mientras se mueve. Las trayectorias ayudan al robot a moverse de manera segura y eficiente. Las velocidades angulares y lineales son clave para determinar cómo se mueve el robot. La velocidad angular afecta la rapidez con la que el robot puede girar, mientras que la velocidad lineal influye en la velocidad a la que se desplaza. Estas velocidades son importantes para seguir trayectorias suaves o hacer movimientos rápidos según lo que se necesite.

Análisis de los códigos:

El código es una simulación que representa el movimiento de un robot en un entorno virtual. Utilizando velocidades de referencia y técnicas de integración numérica, el código actualiza continuamente la posición y orientación del robot en función de estas velocidades y el tiempo de simulación. También se muestra una representación gráfica en 3D que revela la trayectoria seguida por el robot mientras se mueve en el entorno virtual.

Las velocidades de referencia pueden ser lineales o angulares y se utilizan para controlar el tipo de movimiento del robot. Al ajustar estas velocidades, el robot puede avanzar en línea recta, girar en el lugar o realizar movimientos más complejos según sea necesario para la simulación.

Dentro del bucle de simulación, se calcula la nueva posición y orientación del robot en cada iteración. Esto se logra mediante la integración numérica, que tiene en cuenta las velocidades de referencia y el intervalo de tiempo para determinar cómo se actualizan las coordenadas del robot.

La representación gráfica en 3D brinda una visualización clara de la trayectoria del robot, lo que permite evaluar su rendimiento y comprender mejor su movimiento en el entorno virtual.

Código utilizado:

%Limpieza de pantalla

clear all

close all

clc

%%%

TIEMPO

%%%

tf=8; % Tiempo de simulación en segundos (s)

ts=0.1; % Tiempo de muestreo en segundos (s)

t=0:ts:tf; % Vector de tiempo

cont=0;

N= length(t); % Muestras

%%%

CONDICIONES

INICIALES

```

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%Inicializamos las variables que se van a emplear
x1= zeros (1, N+1);      % Posición (X) en el centro del eje que une las ruedas en metros
(m)
y1= zeros (1, N+1);      % Posición (Y) en el centro del eje que une las ruedas en metros
(m)
phi= zeros (1, N+1);     % Orientación del robot en radiaanes (rad)
%Damos valores a nuestro punto inicial de posición y orientación
x1(1)=0; %Posición inicial eje x
y1(1)=0; %Posición inicial eje y
phi(1)=0; %Orientación inicial del robot
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% PUNTO DE CONTROL
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%Inicializamos el punto de control
hx= zeros (1, N+1);      % Posición en el eje (X) del punto de control en metros (m)
hy= zeros (1, N+1);      % Posición en el eje (Y) del punto de control en metros (m)
%Igualamos el punto de control con las proyecciones X1 y Y1 por su
%coincidencia
hx(1)= x1(1);           % Posición del punto de control en el eje (X) metros (m)
hy(1)= y1(1);           % Posición del punto de control en el eje (Y) metros (m)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% VELOCIDADES DE REFERENCIA
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%function [v_r,w_r] = rotacion(N)
%v = 0*ones(1,N); % Velocidad lineal de referencia (m/s)
%w = pi/4*ones(1,N); % Velocidad angular de referencia (rad/s)
%end
%function [v_d,w_d] = dezplasamiento(N)
%v = 1*ones(1,N); % Velocidad lineal de referencia (m/s)
%w = 0*ones(1,N); % Velocidad angular de referencia (rad/s)
%end
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% BUCLE DE SIMULACION
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
for k=1:N
    % Definir la variable accion para seleccionar la acción correspondiente
    if k < N/tf || 2*N/tf <= k && k < 3*N/tf || 4*N/tf <= k && k < 5*N/tf || 6*N/tf <= k && k < 7*N/tf
    || 8*N/tf <= k && k < 9*N/tf
        accion = 1; % Avanzar
    elseif N/tf <= k && k < 2*N/tf || 3*N/tf <= k && k < 4*N/tf || 5*N/tf <= k && k < 6*N/tf || 7*N/tf
    <= k && k < 8*N/tf
        accion = 2; % Rotar
    end

    % Seleccionar la velocidad lineal y angular correspondiente según la acción
    if accion == 1 % Avanzar
        v = 1*ones(1,N); % Velocidad lineal de referencia (m/s)
        w = 0*ones(1,N); % Velocidad angular de referencia (rad/s)
    elseif accion == 2 % Rotar
        v = 0*ones(1,N); % Velocidad lineal de referencia (m/s)

```

```

w = pi/2*ones(1,N); % Velocidad angular de referencia (rad/s)
end

%Aplico la integral a la velocidad angular para obtener el angulo "phi" de la orientación
phi(k+1)=phi(k)+w(k)*ts; % Integral numérica (método de Euler)

xp1=v(k)*cos(phi(k));
yp1=v(k)*sin(phi(k));
%Aplico la integral a la velocidad lineal para obtener las cordenadas
%"x1" y "y1" de la posición
x1(k+1)=x1(k)+ ts*xp1; % Integral numérica (método de Euler)
y1(k+1)=y1(k)+ ts*yp1; % Integral numérica (método de Euler)
% Posicion del robot con respecto al punto de control
hx(k+1)=x1(k+1);
hy(k+1)=y1(k+1);
end

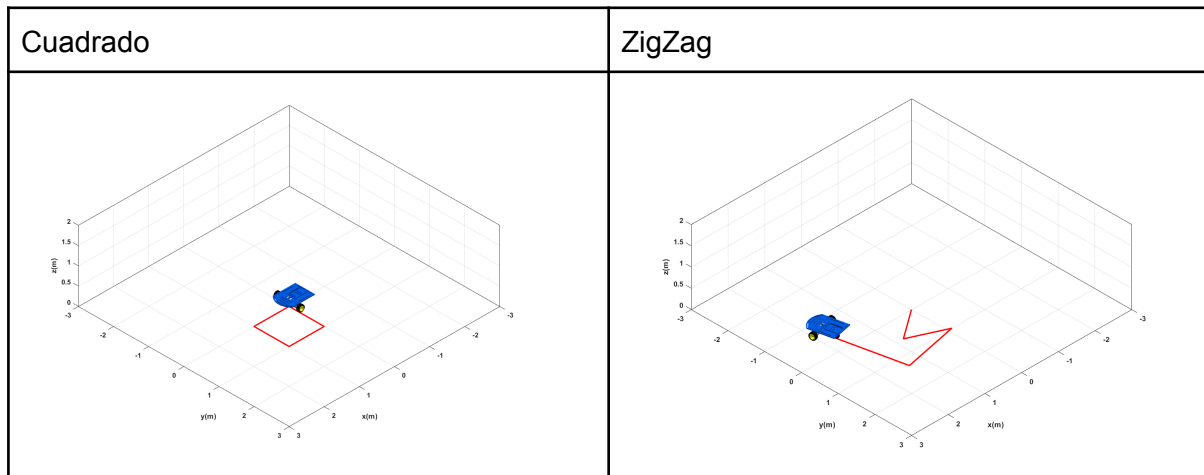
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% SIMULACION VIRTUAL 3D
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% a) Configuracion de escena
scene=figure; % Crear figura (Escena)
set(scene,'Color','white'); % Color del fondo de la escena
set(gca,'FontWeight','bold'); % Negrilla en los ejes y etiquetas
sizeScreen=get(0,'ScreenSize'); % Retorna el tamaño de la pantalla del computador
set(scene,'position',sizeScreen); % Configurar tamaño de la figura
camlight('headlight'); % Luz para la escena
axis equal; % Establece la relación de aspecto para que las unidades de datos sean las
mismas en todas las direcciones.
grid on; % Mostrar líneas de cuadrícula en los ejes
box on; % Mostrar contorno de ejes
xlabel('x(m)'); ylabel('y(m)'); zlabel('z(m)'); % Etiqueta de los eje
view([135 35]); % Orientacion de la figura
axis([-3 3 -3 3 0 2]); % Ingresar limites minimos y maximos en los ejes x y z [minX maxX
minY maxY minZ maxZ]
% b) Graficar robots en la posicion inicial
scale = 4;
MobileRobot;
H1=MobilePlot(x1(1),y1(1),phi(1),scale);hold on;
% c) Graficar Trayectorias
H2=plot3(hx(1),hy(1),0,'r','LineWidth',2);
% d) Bucle de simulacion de movimiento del robot
step=1; % pasos para simulacion
for k=1:step:N
    delete(H1);
    delete(H2);

    H1=MobilePlot(x1(k),y1(k),phi(k),scale);
    H2=plot3(hx(1:k),hy(1:k),zeros(1,k),'r','LineWidth',2);

```

```
pause(ts);  
end
```

Resultados:



Conclusiones:

En conclusión, en el campo de la robótica, las trayectorias desempeñan un papel crucial al permitir que los robots se muevan de manera fluida y segura. Las velocidades angulares y lineales son fundamentales para determinar cómo se sigue una trayectoria. Este código en particular simula el movimiento de un robot en un entorno virtual, utilizando velocidades de referencia para controlar si el robot avanza o rota. A través de integrales numéricas, se actualiza la posición y orientación del robot en cada paso de tiempo. Todo esto se visualiza en una representación gráfica en 3D, lo que facilita la comprensión y observación del movimiento del robot y su trayectoria. En resumen, este código proporciona una herramienta para simular y visualizar el movimiento de robots de una manera accesible y comprensible.